



“Responsabilidad con pensamiento positivo”

UNIVERSIDAD TECNOLÓGICA ISRAEL

**TRABAJO DE TITULACIÓN EN OPCIÓN AL GRADO DE:
INGENIERO EN ELECTRÓNICA DIGITAL Y
TELECOMUNICACIONES**

TEMA:

**DISPOSITIVO ELECTRÓNICO PARA LA CONVERSIÓN DE UN
PROYECTOR ESTÁNDAR EN INTERACTIVO**

AUTOR:

JOSÉ ADID RONQUILLO LUGO

TUTOR:

Ing. Luis H. Montoya L., Msc.

QUITO, ECUADOR

2018

UNIVERSIDAD TECNOLÓGICA ISRAEL

APROBACIÓN DEL TUTOR

En mi calidad de tutor del trabajo de titulación certifico:

Que el trabajo de titulación **“DISPOSITIVO ELECTRÓNICO PARA LA CONVERSIÓN DE UN PROYECTOR ESTÁNDAR EN INTERACTIVO”**, presentado por el Señor Jose Adid Ronquillo Lugo, estudiante de la carrera de Electrónica Digital y Telecomunicaciones, reúne los requisitos y méritos suficientes para ser sometido a la evaluación del Tribunal de Grado, que se designe, para su correspondiente estudio y calificación.

Quito D.M. 15 de Agosto del 2018

TUTOR

Ing. Luis H. Montoya L., Msc.

AGRADECIMIENTOS

Quiero agradecer a todas las personas que hicieron posible esta investigación, y que de alguna manera estuvieron presentes.

A mi familia José, Luz Marina, Manuel y Daniel, que me apoyaron durante todos mis estudios; a Adriana, mi mujer, que me ha apoyado y soportado en todo el desarrollo del final de mi carrera y ha puesto en desorden mi vida.

A Patricio M., Iván B., Mauricio Ch. y Cristian P., compañeros de universidad, con los que hemos compartido varios momentos durante estos años de estudio.

A Winilber P., cuya ayuda (aunque impensada) fue el inicio de mi carrera en Electrónica (y olvide mencionarlo en mi grado del colegio).

A Mauricio O., cuyo apoyo también fue fundamental para poder retomar los estudios y terminar hoy la carrera.

A mis profesores de la Universidad Tecnológica Israel que con su experiencia y conocimientos me orientaron en el desarrollo de mi carrera y en el feliz término de este trabajo.

DEDICATORIA

Este trabajo, para mí, implica el final de una carrera que fue iniciada hace mucho tiempo, dedico este trabajo a las personas más cercanas de mi vida, que sin ellos no sería lo mismo esforzarse para avanzar: mi papa, mi mama, mis hermanos, mi mujer. ¡Sin ustedes no podría avanzar!

ÍNDICE DE CONTENIDO:

APROBACIÓN DEL TUTOR	i
AGRADECIMIENTOS	ii
DEDICATORIA	iii
RESUMEN	vii
ABSTRACT	viii
INTRODUCCIÓN	1
ANTECEDENTES	2
ALCANCE	4
OBJETIVOS	5
DESCRIPCIÓN DEL CONTENIDO DE LOS CAPÍTULOS	6
CAPÍTULO 1	7
FUNDAMENTACIÓN TEÓRICA	7
1.1: MARCO TEÓRICO Y CONCEPTUAL	7
1.2.: MICROCONTROLADOR	7
1.3: DISPOSITIVOS DE ENTRADA (MÓDULO DE POSICIONAMIENTO INFRARROJO)	10
1.4: DISPOSITIVOS DE SALIDA – INTERCONEXIÓN CON LA COMPUTADORA	12
CAPÍTULO 2	13
PROPUESTA	13
2.1: DIAGRAMA ESQUEMÁTICO DEL DISPOSITIVO	13
2.2: ALGORITMO DE PROCESAMIENTO DE LA SEÑAL	17
CAPÍTULO 3	20
DESARROLLO	20
3.1: IMPLEMENTACIÓN	21
3.1: PRUEBAS	26

3.2: ANÁLISIS DE RESULTADOS	29
CONCLUSIONES.....	31
RECOMENDACIONES	32
BIBLIOGRAFÍA	33
ANEXO 1	34
MANUAL TÉCNICO DEL DISPOSITIVO.....	34
ANEXO 2. ALGORITMO DEL PROGRAMA.....	41
ANEXO 3. CRONOGRAMA DE DESARROLLO DEL PROYECTO.....	44
ANEXO 4: HOJA DE ESPECIFICACIONES TÉCNICAS PROCESADOR ATMEGA328	45
ANEXO 5. HOJA DE ESPECIFICACIONES TÉCNICAS PROCESADOR ATMEGA32U4	46

ÍNDICE DE TABLAS Y FIGURAS:

Figura. 1. (1a) Proyección con pantalla (1b) Proyección sin pantalla (con brillo).....	2
Tabla. 1. 1. Cuadro comparativo entre microcontroladores	8
Tabla. 1. 2. Características del Arduino Pro Mini	8
Tabla. 1. 3. Características del Arduino Pro Micro	9
Tabla. 1. 4. Características del módulo de seguimiento infrarrojo.....	11
Figura. 2. 2. Diagrama de bloques del prototipo	13
Figura. 2. 3. Detalle del módulo de posicionamiento.....	13
Figura. 2. 4. Diagrama de pines del Arduino Pro Mini	14
Figura. 2. 5. Diagrama de pines del Arduino Pro Micro	15
Figura. 2. 6. Diagrama esquemático del prototipo	16
Figura. 2. 7. Diagrama esquemático del lápiz infrarrojo	16
Figura. 2. 8. Diagrama de flujo del algoritmo	18
Figura. 3. 9. Dispositivo instalado junto al proyector	20
Figura. 3. 10. Diagrama de conexión entre Arduinos para programación de bootloader....	21
Figura. 3. 11. Imagen del cálculo del fusible del Arduino Pro Mini	22
Figura. 3. 12. Configuración de fusibles del ATmega 328P.....	22
Figura. 3. 13. Placa de circuito impreso vista desde el lado de los componentes	23
Figura. 3. 14. Placa de circuito impreso vista desde el lado cobre.....	24
Figura. 3. 15. Placa de circuito impreso (lado cobre) con los componentes soldados	24
Figura. 3. 16. Placa de circuito impreso vista desde el lado de los componentes	25
Figura. 3. 17. Punta del marcador, se observa el interruptor pulsador y el LED IR	25
Figura. 3. 18. Modo de uso del lápiz IR	26
Figura. 3. 19. Ventana del puerto serie en ausencia de luz infrarroja	27
Figura. 3. 20. Ventana de posición con una fuente de luz infrarroja.....	27
Figura. 3. 21. Ventana de Processing	28

RESUMEN

En este proyecto se diseñó e implementó un prototipo electrónico de bajo costo para lograr la interacción con proyectores. Básicamente un sistema interactivo es un sistema de seguimiento de una fuente de luz (normalmente un lápiz infrarrojo) el cual es relacionado con la posición del cursor del ratón del sistema operativo del computador desde el cual se envía la información, es decir, se logra la interacción con una imagen proyectada “simulando” la posición del ratón del sistema operativo en base a la posición de un lápiz infrarrojo en la pantalla. Generalmente la interacción con proyectores es un accesorio costoso y específico para la marca del equipo.

Con la facilidad de desarrollo de prototipos en plataformas abiertas (PICs, Arduino, Raspberry Pi, etc.) la construcción de un dispositivo que “ubique” una fuente de luz infrarroja y la relacione con la posición relativa al respecto de la pantalla es sencillo de realizar empleando componentes de fácil adquisición y de forma local.

En este proyecto se presenta el diseño e implementación de un prototipo de un sistema de seguimiento de luz infrarroja capaz de enviar la información de ubicación relativa a la pantalla. Esta información es interpretada por el sistema operativo de un computador como la posición del ratón, logrando así la interactividad con contenido proyectado.

Palabras clave:

Posicionamiento

Infrarrojo

Ratón

Interacción

Computador

ABSTRACT

In this project a low-cost prototype for whiteboard type interaction with a projected screen was designed and the built. Basically, an interactive projector uses a light source (typically an infrared pen) as an input to calculate its relative position to the projected image. It uses the relative position of the infrared pen to simulate the mouse's cursor position by Operating System and so controlling how the mouse moves (and clicks). Usually, this is commercially done with expensive equipment which typically is vendor locked.

With open platforms such as Arduino, Microchip PIC, Raspberry Pi, etc. building a device capable of "finding" the relative coordinates (x, y) of a light source, process and send them to the computer can be achieved easily with low cost components.

In this project the designing and building of the device is discussed, along with the theoretical basis and algorithm development.

Keywords:

Interactive

Infrared

Positioning

Mouse

INTRODUCCIÓN

Actualmente, el hombre está acostumbrado a interactuar con sus dispositivos electrónicos en prácticamente todos los aspectos de la vida. El uso de los teléfonos inteligentes con pantalla táctil y el uso de tecnologías “touch” en otros dispositivos obligan a buscar un uso sencillo e intuitivo en casi todas las actividades cotidianas.

Una de las actividades, que todavía no ha “migrado” extensamente a esta interacción, es la proyección de diapositivas o de información de texto. Para interactuar en dichas ocasiones hay varias soluciones en el mercado cada una con sus pro y contras, por ejemplo: pantallas táctiles – Microsoft Surface Hub, proyectores interactivos – NEC, Epson, Optoma, dispositivos de interfaz – Tablet Wacom, entre otras.

Entre las ventajas de usar un proyector diseñado para la interacción es que se cuenta con soporte de fábrica, stock de repuestos, actualizaciones de software. Las desventajas de dichos sistemas propietarios suelen ser su alto costo y que no existe compatibilidad entre marcas, es decir, si se adquiere un proyector marca A los accesorios de marca B no son compatibles.

Los monitores de pantalla táctil son de los más costosos y limitados en tamaño. Pues usan dispositivos infrarrojos para detección o son construidos con técnicas similares a las usadas en teléfonos móviles (pantalla táctil resistiva o capacitiva).

Las Tablet digitalizadoras (tipo Wacom) son dispositivos diseñados para “dibujar” a mano en el PC, permitiendo la interacción de diseñadores gráficos con programas especializados, además de que requieren controladores específicos.

Otra solución son los proyectores interactivos, los cuales usan tecnología propietaria de cada fabricante que puede ser infrarroja, bluetooth, radiofrecuencia, etc. que son únicamente compatibles con equipos del mismo fabricante.

El sistema operativo de una computadora es capaz de procesar la información de coordenadas enviadas por un ratón (mouse) y desplegar la posición relativa del cursor basado en dichas coordenadas. Usando un módulo de posicionamiento infrarroja capaz de ubicar, en coordenadas relativas de posición, una fuente de luz infrarroja, procesando dichas coordenadas de manera que el sistema operativo sea capaz de interpretarlas a manera de movimiento de ratón.

ANTECEDENTES

En múltiples ambientes (comerciales, educativos y similares) donde se requiere presentar información a una audiencia de varias personas normalmente se proyecta el contenido, el cual puede ser multimedios, diapositivas, documentos, etc.

En más de una ocasión es necesario interactuar con dicho contenido: hacer anotaciones, resaltar alguna idea o simplemente “dibujar” algo en la proyección para aclarar la idea presentada. Generalmente cuando esto es requerido, dependiendo del tipo de superficie o pantalla donde esté siendo proyectada la imagen, se usan marcadores de tinta, lápices o dispositivos interactivos que no rayan la superficie.

En general, la mayoría de los proyectores son tradicionales; es decir, no soportan dispositivos interactivos, lo cual implica que no es posible interactuar de manera eficiente con la información proyectada. A veces, el presentador prefiere proyectar la imagen sobre una pizarra para poder interactuar con lo presentado. Pero esto no es óptimo pues la pizarra tiene un brillo propio que distorsiona la presentación a una parte de la audiencia o simplemente el marcador empleado es de un color que no es posible observar por ciertos miembros de la audiencia, dañando el objetivo de interactuar del presentador. En la figura 0.1. abajo se puede observar que se debe usar la pantalla de proyección (Figura 1a) para evitar el brillo o la proyección con brillo (Figura 1b) – cuando no se usa la pantalla – que impide que todo el contenido se pueda observar correctamente.

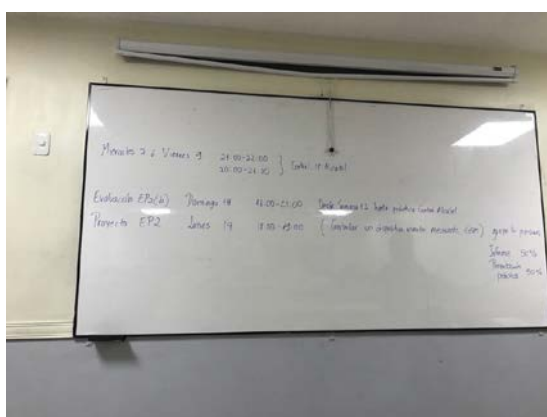


Figura. 1. (1a) Proyección con pantalla (1b) Proyección sin pantalla (con brillo)

Fuente: Elaborado por el autor

Usar un proyector interactivo es una gran solución para cuando es requerida la interacción con la información proyectada, con el limitante de que generalmente es costoso y requiere equipo especializado, algunas veces incluso con software propio del fabricante. Lo que limita su uso a ambientes muy específicos.

En varios centros educativos del país existen proyectores en las aulas, pero dichos proyectores, al ser estándar no permiten interacción alguna. Reemplazar dichos equipos por proyectores interactivos tendría un costo muy alto, pues se debe prorratear el costo del proyector, el software, los insumos, instalación, calibración, etc. Eso puede dejar fuera del presupuesto de la mayoría de las instituciones usar proyectores interactivos.

Un sencillo dispositivo electrónico capaz de ubicar una fuente de luz infrarroja es una solución a los inconvenientes de interacción en proyectores estándar, pues permite de una manera sencilla, económica y no especializada (no requiere la instalación de software) de interactuar con la información presentada, facilitando la presentación.

Este proyecto incluye el diseño y construcción de un prototipo de dispositivo capaz de realizar la ubicación relativa de una fuente de luz al respecto de la información proyectada, procesar dicha ubicación y enviarla como un dispositivo de interfaz humana (*Human Interface Device*) a un computador – independiente del sistema operativo – para lograr interacción con cualquier tipo de proyector.

El dispositivo usará programación con software abierto, por lo que será posible realizar cualquier tipo de programa orientado a mejorar la experiencia de aprendizaje y explicación en clases o cualquier otro uso que se le quiera dar.

ALCANCE

En el presente proyecto se plantea las bases teóricas del funcionamiento del prototipo del dispositivo, el algoritmo para lograr una funcionalidad básica basada en simular el movimiento del cursor del ratón de un computador usando un lápiz infrarrojo y una manera sencilla de implementarlo con componentes de fácil adquisición en el mercado nacional. Funciones avanzadas, tales como: aplicaciones de software para PC, calibración, proyección o diseño de monturas a la medida para la instalación no serán realizadas.

Al ser un prototipo el dispositivo será sencillo de implementar y podrá ser usado como base para el desarrollo de un dispositivo comercial más elaborado, con manufactura profesional e interfaz de calibración para computador.

OBJETIVOS

Objetivo general:

Crear un dispositivo para transformar en interactivo un proyector normal.

Objetivos específicos:

- Definir los elementos constitutivos de un dispositivo electrónico para transformar un proyector estándar en interactivo.
- Elaborar el diagrama de bloques para el funcionamiento del dispositivo.
- Diseñar el circuito electrónico para la implementación del dispositivo.
- Desarrollar el algoritmo de programación para el funcionamiento del dispositivo.
- Implementar el dispositivo.
- Realizar la validación del dispositivo a través de pruebas de uso.

DESCRIPCIÓN DEL CONTENIDO DE LOS CAPÍTULOS

Capítulo 1: en este capítulo se presenta el marco teórico relacionado con el dispositivo desde el punto de vista para su elaboración. No se incluirán detalles de qué es la luz infrarroja, cómo funciona un ratón, qué es un USB y similares, pues se asume que el lector está familiarizado con las tecnologías empleadas y más bien, se hará una breve explicación de los principios sobre los cuales funciona el dispositivo.

Capítulo 2: en este capítulo se evalúan las opciones de diseño e implementación del prototipo, se analiza brevemente los componentes y las opciones para su construcción.

Capítulo 3: en este capítulo se documenta el proceso de construcción empleado, así como el algoritmo de programación desarrollado para lograr la operación de este. Así mismo se resume las pruebas realizadas de operación, y se documentan las experiencias de implementación.

Conclusiones y recomendaciones: como su nombre lo indica se presentan los resultados finales del proyecto, los objetivos alcanzados y algunas recomendaciones para mejora y trabajos futuros.

CAPÍTULO 1

FUNDAMENTACIÓN TEÓRICA

1.1: MARCO TEÓRICO Y CONCEPTUAL

En este capítulo se revisan las tecnologías que permitieron implementar el dispositivo para lograr la interactividad con un proyector, con esta información luego se continúa con el diseño e implementación de este.

Dado que el sistema está basado en microcontrolador primero se analiza el tipo de microcontrolador usado, para luego revisar dispositivos de entrada de información y de salida.

1.2.: MICROCONTROLADOR

Un microcontrolador es básicamente un computador completo, pues incluye:

1. Entradas
2. Unidad de procesamiento o CPU
3. Memoria RAM y ROM
4. Salidas

Los microcontroladores son circuitos integrados capaces de ser programados, incluyen todo lo requerido para ejecutar su programa en una sola placa electrónica. Existen varios tipos de dispositivos microcontroladores diseñados para múltiples funciones y también de uso general.

Dada la gran variedad de dispositivos en el mercado, y teniendo en cuenta sus características, una de las plataformas más populares para implementación de prototipos es Arduino. Este dispositivo surgió del movimiento de “*open-software*” y “*open-hardware*”. Fue creado inicialmente en Italia alrededor del año 2005. En la tabla 1.1 se observa una breve comparación entre controladores.

Tabla. 1. 1. Cuadro comparativo entre microcontroladores

	PIC Microchip	Arduino
Tiene ambiente propio de programación	Si	Si
Se puede programar múltiples veces	Si	Si
Permite programación sin removerlo de la placa	No	Si
Tiene puerto serie embebido, no requiere de hardware extra para operación	No	Si
Existen periféricos compatibles entre sí	Si	Si

Fuente: Elaborado por el autor

Arduino está basado en distintos tipos de microcontrolador (depende del modelo y características a usar), aunque generalmente se usa el ATmega. Existen varios tipos de placas Arduino, cada una diseñada para diversos usos, puntualmente – para desarrollo de este proyecto – serán empleados dos tipos:

1. Arduino Pro Mini
2. Arduino Pro Micro

¿Por qué usar dos microcontroladores para un proyecto? Pues bien, se revisa primero las características de cada uno:

Arduino Pro Mini:

Este dispositivo está fabricado en una placa pequeña, está orientado a proyectos donde el espacio es imperativo y consta de un microprocesador ATmega328 (la hoja de especificaciones técnicas está en el Anexo 3). Existen dos versiones: una que trabaja con 5 voltios y resonador de 16 MHz y una de 3,3 voltios y resonador de 8 MHz. Las especificaciones técnicas de este equipo se detallan en la tabla 1.2.:

Tabla. 1. 2. Características del Arduino Pro Mini

Microprocesador	ATmega 328
Fuente de poder	3,35 a 12v (el modelo de 3,3v) y de 5 a 12 v (el modelo de 5v)
Voltaje de operación del circuito	3,3 a 5v (depende del modelo)
Pines de entrada/salida digital	14
Pines PWM	6

UART	1
SPI	1
I ² C	1
Pines de entrada analógica	6
Interrupciones externas	2
Corriente máxima por pin I/O	40 mA
Memoria Flash	32 KB, (2KB son usados por bootloader)
SRAM	2 KB
EEPROM	1 KB
Velocidad del reloj	8 MHz (la versión de 3,3v) y 16 MHz (la versión de 5v)

Fuente: Elaborado por el autor

Si se observan las características del dispositivo se puede notar que no cuenta con puerto de comunicación serie a bordo, por lo que es necesario usar una tarjeta adicional para poderlo programar de manera sencilla.

Arduino Pro Micro:

Este dispositivo es muy similar al Arduino Pro Mini, de igual manera está fabricado en una placa pequeña y la gran diferencia (con el Pro Mini) es que usa un microprocesador ATmega32U4 que si tiene puerto serie embebido (la hoja de especificaciones técnicas está en el Anexo 4). De igual manera que el Pro Mini, existen dos versiones: una que trabaja con 5 voltios y resonador de 16 MHz y una de 3,3 voltios y resonador de 8 MHz. Las especificaciones técnicas de este equipo se detallan en la tabla 1.3.:

Tabla. 1. 3. Características del Arduino Pro Micro

Microprocesador	ATMega 32U4
Fuente de poder	4 a 12v (el modelo de 3,3v) y de 5 a 12 v (el modelo de 5v)
Voltaje de operación del circuito	3,3 a 5v (depende del modelo)
Pines de entrada/salida digital	12
Pines PWM	5
Puerto de comunicación	1 Serie Tx-Rx en hardware con puerto micro-USB

SPI	1
I ² C	1
Pines de entrada analógica	4
Interrupciones externas	2
Corriente máxima por pin I/O	40 mA
Memoria Flash	32 KB, (2KB son usados por bootloader)
SRAM	2,5 KB
EEPROM	1 KB
Velocidad del reloj	8 MHz (la versión de 3,3v) y 16 MHz (la versión de 5v)

Fuente: Elaborado por el autor

Este dispositivo cuenta con un puerto micro-USB implementado en la placa, facilitando así la programación y uso del dispositivo.

Ambos dispositivos son muy similares físicamente, tienen un diseño pequeño como para ser implementados en una placa de circuito impreso pequeña, la gran diferencia entre los dos es que el Pro Mini no tiene puerto de comunicación en la placa y el Pro Micro sí.

1.3: DISPOSITIVOS DE ENTRADA (MÓDULO DE POSICIONAMIENTO INFRARROJO)

Para lograr la interactividad se usan dos dispositivos como entrada del sistema. Ambos son dispositivos digitales, un módulo de posicionamiento y un lápiz infrarrojos. El diagrama de bloques del dispositivo se observa en la figura 2.2 en el capítulo dos.

Módulo de posicionamiento infrarrojo:

Este dispositivo es la parte más importante del proyecto, pues, es el que realiza el seguimiento y envía la información de posicionamiento de la fuente de luz infrarroja para su posterior procesamiento en el Arduino Pro Micro.

El módulo básicamente es un dispositivo de posicionamiento IR capaz de hacer el seguimiento y posicionamiento de hasta cuatro fuentes de luz infrarroja de manera simultánea, enviando la información de ubicación relativa (coordenadas x, y) y el tamaño circular (*blob*) de la fuente infrarroja. Esto puede ser usado en caso de que se desee usar más

de un lápiz infrarrojo de manera simultánea. Pero, por limitaciones del sistema operativo, no es recomendable usar más de un ratón al tiempo.

Este dispositivo contiene un módulo de semiconductor CMOS que cuenta con un filtro pasa-banda infrarrojo (de allí que solo es “sensible” a la luz infrarroja), cuenta con DSP (*Digital Signal Processor*) y entrega la información digital en protocolo I²C para su procesamiento.

En la tabla 1.4., se muestran algunas especificaciones técnicas de este dispositivo (para obtenerlas todas el fabricante - *PixArt* - requiere una compra inicial de 1000 unidades y firmar un acuerdo de confidencialidad, por lo que únicamente están disponibles algunas):

Tabla. 1. 4. Características del módulo de seguimiento infrarrojo

Sensor de imagen	CMOS
Voltaje de operación	2,0 a 3,6v
Consumo máximo de corriente	44 mA
Reloj del sistema	De 5 a 23 MHz
Interfaz	I ² C
Resolución	128 x 96 pixeles
Número de objetos a seguir	Máximo 4 simultáneos
Capaz de entregar información de posición	Sí, mediante I ² C en coordenadas X y Y
Ángulo de visión vertical	23°
Ángulo de visión horizontal	33°
Distancia de detección	0 a 3 metros

Fuente: Elaborado por el autor

El protocolo mediante el cual este módulo envía los datos es I²C o *Inter-Integrated Circuit* (Entre circuitos integrados) el cual es un bus serie inventado por Philips Semiconductor en 1982.

El protocolo fue diseñado inicialmente para permitir el envío de datos entre periféricos de baja velocidad y microcontroladores en una corta distancia en una placa de circuito impreso.

El protocolo usa dos líneas bidireccionales de datos:

1. Línea de Datos Seriales (*Serial Data Line* o SDA)
2. Línea de Reloj Serie (*Serial Clock Line* o SCL)

Las líneas de datos generalmente funcionan con lógica de +5 ó +3,3 voltios, es de prioridad (un dispositivo es el maestro y los demás esclavos) y usa velocidades de 100 kbit/s. El número total de nodos está limitado por la tabla de direcciones (usa direccionamiento de 7 bits) y la capacitancia total del bus que es de 400 pF. Dicha impedancia implica que se requiere usar una referencia de tierra común en toda la comunicación.

Lápiz infrarrojo:

Este dispositivo es la fuente de luz infrarroja. Es un circuito serie sencillo, con un diodo LED infrarrojo, una resistencia de 220 Ω (usada como limitadora de corriente) un interruptor miniatura y una pila o batería, su diagrama esquemático se observa en la figura 2.7 del Capítulo 2.

Al cerrar el interruptor se activa el circuito encendiendo el LED infrarrojo, generando así la fuente de luz.

1.4: DISPOSITIVOS DE SALIDA – INTERCONEXIÓN CON LA COMPUTADORA

Para la interconexión con la computadora se usa el puerto USB integrado en el Arduino Pro Micro, conectándolo mediante un cable de micro-USB (terminal tipo Micro-B) a USB (Terminal tipo A).

El USB (Bus Serie Universal o *Universal Serial Bus*) es un estándar de la industria de computadores desarrollado para conectar, comunicar y enviar energía entre distintos dispositivos (por ejemplo: la computadora y periféricos).

Es un protocolo de comunicación serial diseñado para facilitar la interconexión de periféricos a la computadora. El sistema consiste en un host con uno o más puertos de comunicación “hacia abajo” o *downstream*, esto forma una configuración de topología tipo estrella, un host USB puede soportar un máximo de 127 dispositivos. El estándar de USB 2.0 permite una distancia máxima entre host de 5 metros si el dispositivo está funcionando a 480 Mbit/s y funciona a 5 voltios con una tolerancia de $\pm 5\%$.

CAPÍTULO 2

PROPUESTA

El prototipo, al ser un dispositivo digital, consta de entradas, procesador y salidas; con un diagrama de bloques como se observa en la figura 2.2:

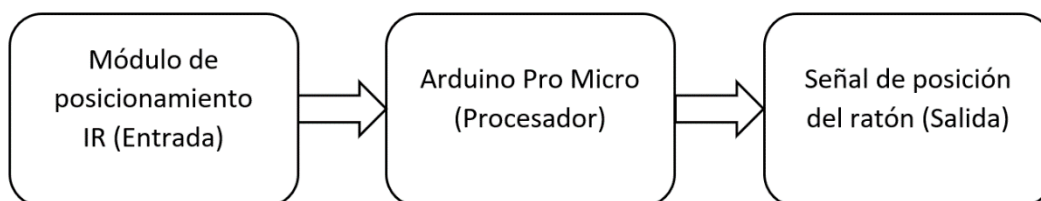


Figura. 2. 2. Diagrama de bloques del prototipo

Fuente: Elaborado por el autor

Así entonces, se desarrolla cada bloque para su posterior construcción, como se explica en las siguientes secciones.

2.1: DIAGRAMA ESQUEMÁTICO DEL DISPOSITIVO

Como entrada para el sistema se emplea el módulo de posicionamiento IR, el cual es un dispositivo que cuenta con 8 pines, según se observa en la figura 2.3:

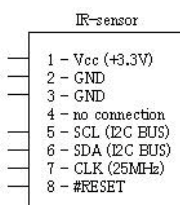


Figura. 2. 3. Detalle del módulo de posicionamiento

Fuente: (Hobley, 2009)

Como se observa, el módulo solamente soporta alimentación de 3,3 voltios; por ende, la línea de datos I²C debe ser también de 3,3 voltios. Es por esto que se emplean varios controladores Arduino de 3,3 voltios para no hacer conversión del nivel de la señal.

Para que el dispositivo funcione se deben tener dos señales:

1. El pin de reset debe estar a nivel uno lógico (1L), lo cual se logra mediante un capacitor de 1 μ F conectado a tierra (que se descargará al apagar el equipo) y una resistencia de 30k Ω usada como limitadora de corriente.

cuales se conectan a los pines correspondientes en el Arduino Pro Micro. Dichos pines corresponden a los pines digitales 2 y 3; como se observa en la figura 2.5.

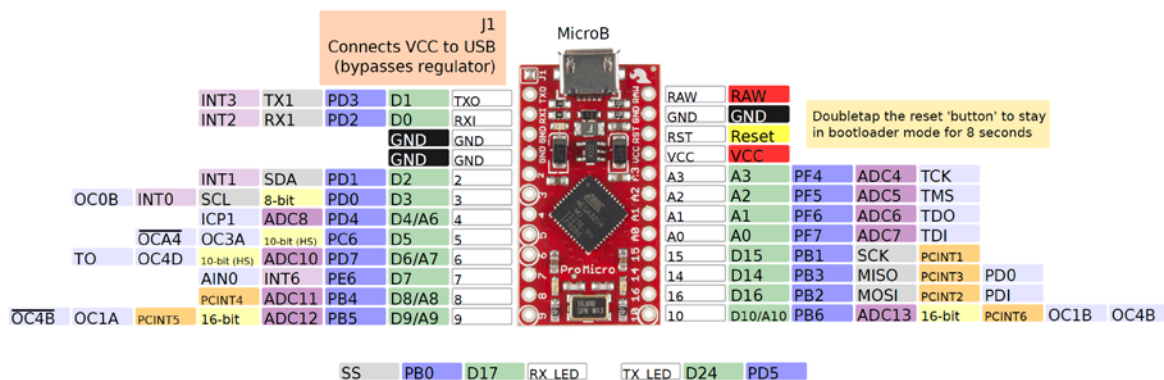


Figura. 2. 5. Diagrama de pines del Arduino Pro Micro

Fuente: (Sparkfun Electronics, 2016)

La fuente de poder de todo el circuito es el Arduino Pro Micro pues toma su energía desde el puerto USB de la computadora y la regula. De aquí se toma la salida de 3,3 voltios para alimentar todos los dispositivos, obteniendo la señal de tierra de referencia (GND).

Debido a que el Arduino Pro Micro requiere un reinicio o “reset” antes de su programación (para ingresar a modo *bootloader*) se usa un pulsador normalmente abierto conectado entre los pines de Reset y Tierra (GND) para facilitar la programación una vez que el dispositivo esté montado en la placa de circuito impreso.

Debido al uso del puerto USB integrado en el Pro Micro no es requerido ningún terminal adicional para conectarlo al computador, lo que sí se requiere es un interruptor para encender y apagar el “modo ratón”. Si dejamos al Arduino en “modo ratón” todo el tiempo luego será muy difícil hacer algún cambio en el programa o mejorar el código, pues siempre aparecerá en la computadora como un ratón.

Para esto se usa un pin de entrada digital que en nivel uno lógico (1L) activa el “modo ratón” y en cero lógico (0L) lo desactiva. Esto se logra con un simple divisor de voltaje con una resistencia de 10KΩ conectado entre GND y el interruptor al pin digital 9 del Arduino Pro Micro.

Con todo lo indicado, el diagrama esquemático completo del dispositivo es como se indica en la figura 2.6.

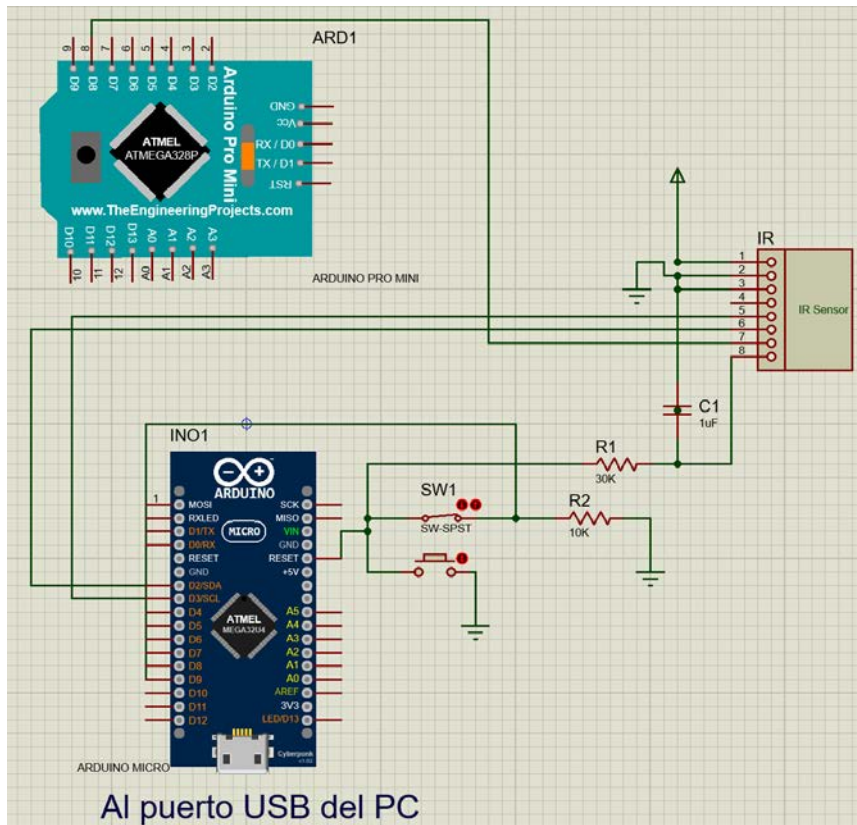


Figura. 2. 6. Diagrama esquemático del prototipo

Fuente: Elaborado por el autor

En el diagrama esquemático se observa en J1 la conexión al módulo de posicionamiento. Como se puede notar, al ser un dispositivo completamente digital, el diagrama esquemático es muy sencillo.

Como fuente de luz infrarroja se emplea un circuito serie montado a manera de lápiz, el esquemático del circuito está indicado en la figura 2.7:

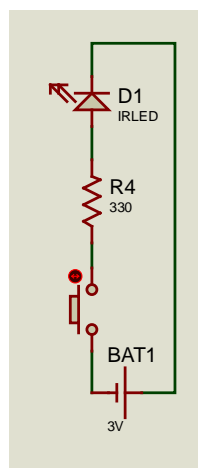


Figura. 2. 7. Diagrama esquemático del lápiz infrarrojo

Fuente: Elaborado por el autor

2.2: ALGORITMO DE PROCESAMIENTO DE LA SEÑAL

Para el desarrollo del algoritmo se utilizó el ambiente de diseño integrado (IDE *Integrated Development Environment*) de Arduino. Este IDE permite el uso del puerto serie del Arduino Pro Micro para confirmar los datos conforme son recibidos. El algoritmo completo está en el Anexo 2.

Para lograr la comunicación I²C de los dispositivos se emplean las siguientes librerías de Arduino:

- Wire.h: que incluye todo lo necesario para la comunicación I²C.
- PVision.h: que incluye la inicialización I²C del módulo y todo lo necesario para recibir correctamente la señal.
- Mouse.h: que permite el modo de emulación de ratón o “modo ratón”.

Estas librerías son públicas (están disponibles en <http://www.arduino.cc> (Arduino, 2018)) y permiten su modificación. Se modificó el algoritmo para para obtener los datos como son requeridos.

El algoritmo del programa se construye en “partes”, siguiendo la estructura de un programa de Arduino; de la siguiente manera:

1. Parte 1: se declaran las variables y se realiza el direccionamiento de la comunicación
2. Parte 2: se configura la comunicación serie y cómo se indicarán los datos
3. Parte 3: se realiza el procesamiento de la señal recibida y se configura el despliegue de la información
4. Parte 4: se envía la información procesada al “modo ratón” para que la computadora la use

El diagrama de flujo que permitió desarrollar el algoritmo se observa en la figura 2.8.

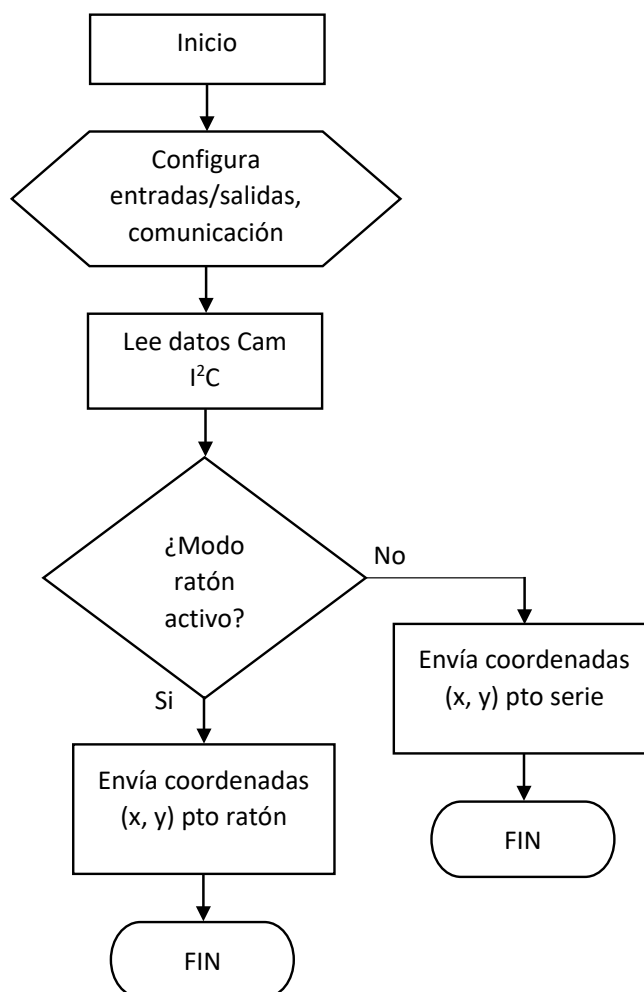


Figura. 2. 8. Diagrama de flujo del algoritmo

Fuente: Elaborado por el autor

Dado que el módulo de posicionamiento es capaz de procesar hasta cuatro fuentes de luz infrarroja de manera simultánea, y no son requeridas tantas fuentes de manera simultánea, se deben eliminar las señales no requeridas de manera de realizar el procesamiento únicamente de la primera fuente.

Entonces se envía o “imprime” la información en el puerto serie, o mejor dicho, programar al sistema para que despliegue la información recibida – que corresponde a las coordenadas y al tamaño de la burbuja o “blob” – en el puerto serie del IDE asignado al Arduino Pro Micro.

Una vez obtenida la información de posición (las coordenadas X y Y) y confirmado que el sistema está enviando información se programa el funcionamiento del ratón, con un pin que lo habilita y lo deshabilita, para escalar las coordenadas recibidas y mover el ratón en base a eso.

Las coordenadas recibidas varían desde 1023 hasta 0; siendo 1023 la no presencia de luz infrarroja y 0 la última coordenada hacia la derecha en el eje X (o hacia arriba en el eje Y); por lo que, se debe escalar dichas coordenadas a 12 niveles de movimiento; sino es hecho este escalamiento el sistema será muy “sensible” a los movimientos del LED infrarrojo y prácticamente imposible de controlar.

Esta información se debe escalar para una resolución de un monitor de 1024 x 768 pixeles (para el eje Y se divide por 1,33 debido a la relación de la resolución de pantalla 4:3); y, dentro de los rangos de movimiento disminuir el campo de acción del puntero.

Otra parte del programa, aprovechando que el módulo es capaz de indicar la presencia de una burbuja (*blob*) y su tamaño, hacemos que el Arduino envíe a la computadora la orden de click izquierdo del ratón “*mouse-click*” de manera de activar la función siempre que el tamaño del blob sea mayor a un valor especificado previamente.

Con esto, únicamente se usa el click izquierdo del ratón, pues para el uso de interactividad no se requieren las funciones de click derecho.

CAPÍTULO 3

DESARROLLO

El uso extenso de microcontroladores ha permitido que muchos de los prototipos sean de tamaño muy pequeño, comparados con el tamaño que tendrían los prototipos si se empleasen dispositivos discretos. Este proyecto no es la excepción, incluso si se usare otro tipo de Arduino (que soporte línea de datos I²C de 3,3v), se podría hacer mucho más pequeño.

La idea de hacer el prototipo pequeño fue debido a que normalmente será instalado junto a un proyector por lo que el espacio y peso son muy importantes. El prototipo montado sobre un proyector se observa en la figura 3.9.



Figura. 3. 9. Dispositivo instalado junto al proyector

Fuente: Elaborado por el autor

3.1: IMPLEMENTACIÓN

Para obtener la señal de reloj del Arduino Pro Mini fue necesario modificar el bootloader del Arduino. Así entonces, se obtiene la señal de reloj del sistema en un pin de conexión.

Para hacer esto se usó un programador para el ATmega, que puede ser otro Arduino, se modifica el archivo “boards.txt” del Arduino IDE para que indique un tipo “nuevo” de

Arduino. Es un tipo “nuevo” pues no es el bootloader estándar de un Arduino. La conexión para esta programación se observa en la figura 3.10, empleando un Arduino Nano como ISP (*In System Programmer* o programador en el sistema) para la programación de cualquier otro Arduino (en el Arduino a programar se conectan las señales RST, MOSI, MISO y SCK en los pines correspondientes):

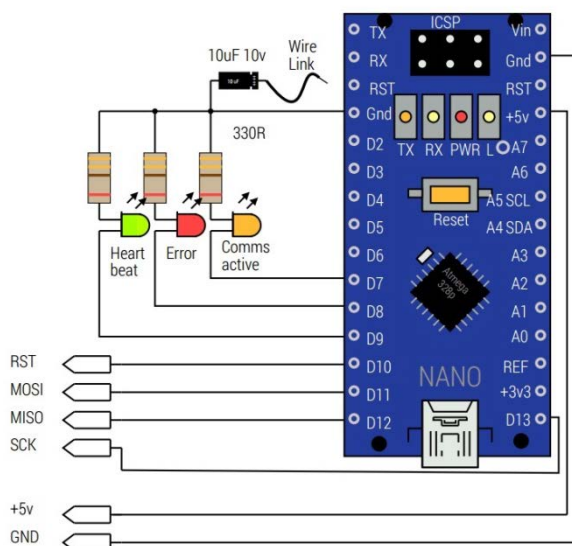


Figura. 3. 10. Diagrama de conexión entre Arduinos para programación de bootloader.

Fuente: (Derwyn, 2017)

Una vez realizadas las conexiones apropiadas (en los pines apropiados en base a la señal) se configuró el software Arduino IDE para que emplee un programador ArduinoISP (esto es, debido a que se dispone de uno), se cambió el bootloader modificando el fusible de configuración que indica SCKO o habilitar la salida de reloj (*clock out*). Para facilitar el cálculo es posible usar una calculadora web (Hämmerling, 2008) (www.engbedded.com/fusecalc) que es una aplicación que calcula los fusibles del ATmega 328P. Una vez el sistema ejecute la programación, tendremos listo el reloj en el pin de salida del Arduino Pro Mini., como se observa en la figura 3.11.

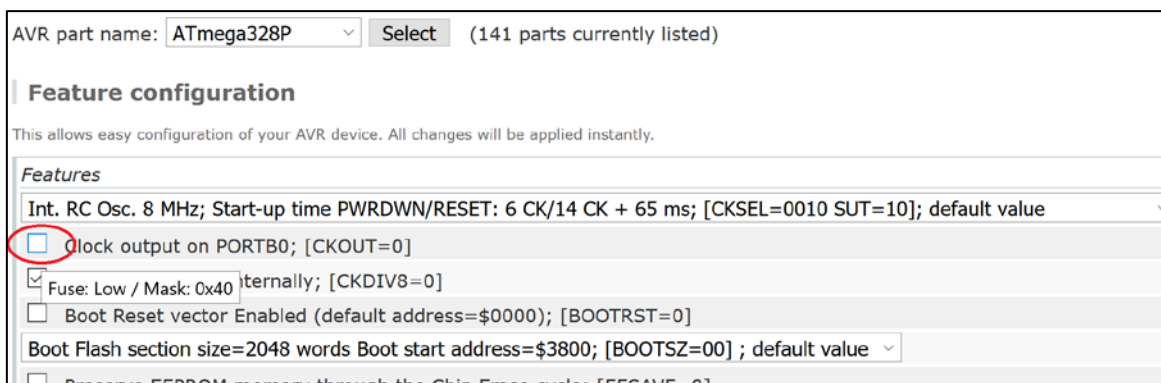


Figura. 3. 11. Imagen del cálculo del fusible del Arduino Pro Mini

Fuente: (Hämmerling, 2008)

Para terminar, se debe modificar el archivo boards.txt (el que lista el tipo de tarjetas usadas) y crear una placa Arduino diferente, por ejemplo: “Arduino Pro Mini con reloj” y en la configuración de los fusibles bajos se debe escribir la opción que nos arroja la página de arriba (0xA2) en lugar de la configuración original 0xFF; dicho cambio se observa en negritas en la figura 3.12.

```

pro328.name=Arduino      Pro      Mini      con      reloj
pro328.upload.protocol=arduino
pro328.upload.maximum_size=30720
pro328.upload.speed=57600
pro328.bootloader.low_fuses=0xA2
pro328.bootloader.high_fuses=0xDA
pro328.bootloader.extended_fuses=0x05
pro328.bootloader.path=atmega
pro328.bootloader.file=ATmegaBOOT_168_atmega328_pro_8MHz.hex
pro328.bootloader.unlock_bits=0x3F
pro328.bootloader.lock_bits=0x0F

pro328.build.mcu=atmega328p
pro328.build.f_cpu=8000000L
pro328.build.core=arduino
pro328.build.variant=standard

```

Figura. 3. 12. Configuración de fusibles del ATmega 328P

Fuente: Elaborado por el autor

Partiendo del diagrama esquemático, y con todos los componentes preparados para la implementación, se procedió a construir la placa de circuito impreso. El método más sencillo para diseño es emplear una aplicación de computadora; pues esta permite automatizar la elección del tipo de línea o pista, revisa los errores u omisiones de conexiones, entre otras funciones. Entonces, para el diseño se tomó en cuenta algunas consideraciones (el cálculo del ancho de las pistas se basa en el estándar general para diseño ANSI-IPC-2221), las cuales son:

- Las líneas de circuito impreso que lleven alimentación (dado el bajo consumo de corriente de los circuitos integrados, y que operará a un máximo de 10°C sobre la temperatura ambiente) deben ser de al menos 0,8 mm.
- Las líneas de señal de circuito impreso son de muy baja corriente por lo que al menos deben ser de 0,5 mm

Esto se hace indicando al sistema de diseño que tipo de línea es, así el software es capaz de elegir el grosor de las líneas y pistas de circuito impreso. El software permite al usuario observar las pistas de circuito impreso desde el lado de los componentes o desde el lado del cobre de la placa, facilitando así el proceso de diseño. Como se observa en la figura 3.13.

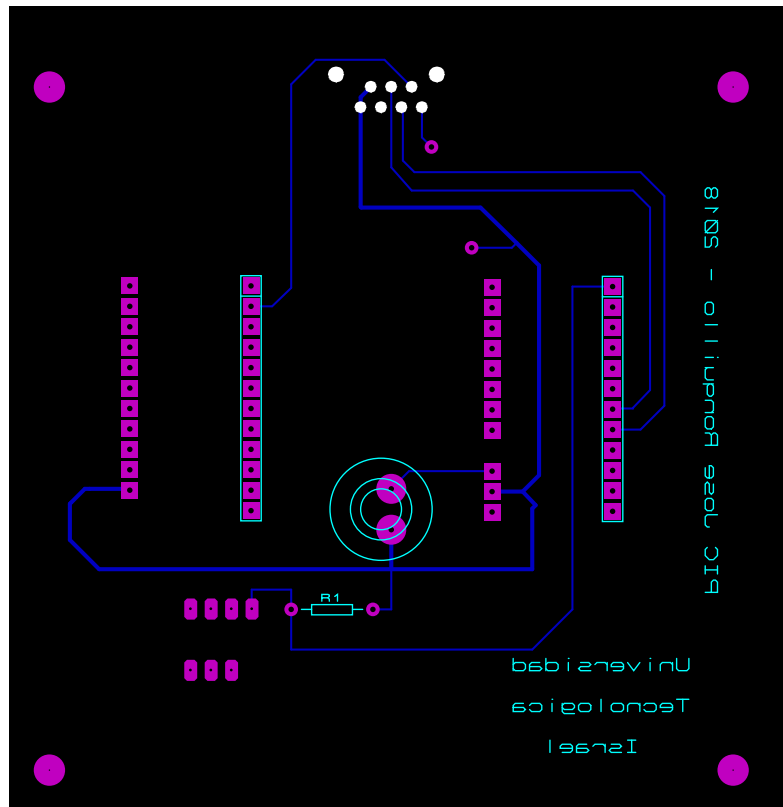


Figura. 3. 13. Placa de circuito impreso vista desde el lado de los componentes

Fuente: Elaborado por el autor

Con esto, al mandar a imprimir la placa se observa como quedan las “pistas” de cobre, en la figura 3.14.

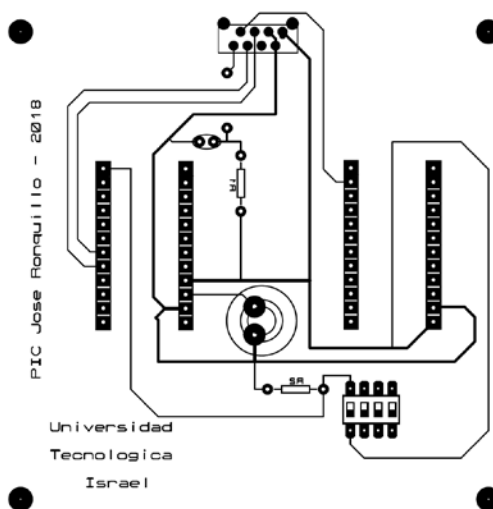


Figura. 3. 14. Placa de circuito impreso vista desde el lado cobre

Fuente: Elaborado por el autor

Usando técnicas de fabricación de placas de circuito impreso (transferencia térmica, con marcador, por CNC, etc.) se fabricó la placa. En este caso la placa es de 10 x 10 cm. De la figura 11 se observa que se usan dos Arduinos, el de la izquierda corresponde al usado como reloj (Arduino Pro Mini) pues solamente tiene una conexión en D8 (y la alimentación eléctrica) y el de la derecha es el procesador de señal (Arduino Pro Micro).

El conector J1 es el conector de la parte de arriba de la placa. Debido a que el módulo es miniatura este conector está planificado para “conectar” con cables terminales a dicho módulo, elevándolo sobre la placa para lograr una mejor visión.

En la figura 3.15 se observa la placa desde el lado de cobre, con los componentes soldados; y, en la figura 3.16 la placa desde el lado de los componentes:

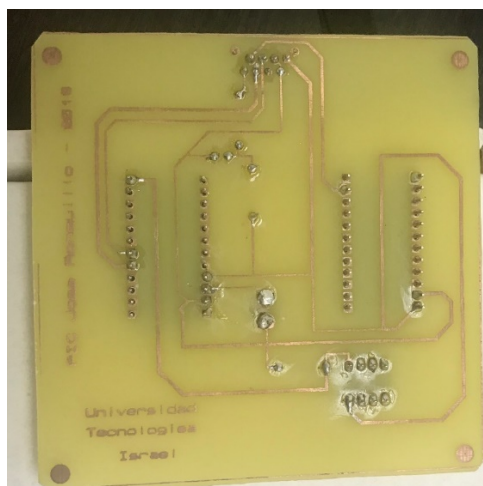


Figura. 3. 15. Placa de circuito impreso (lado cobre) con los componentes soldados

Fuente: Elaborado por el autor

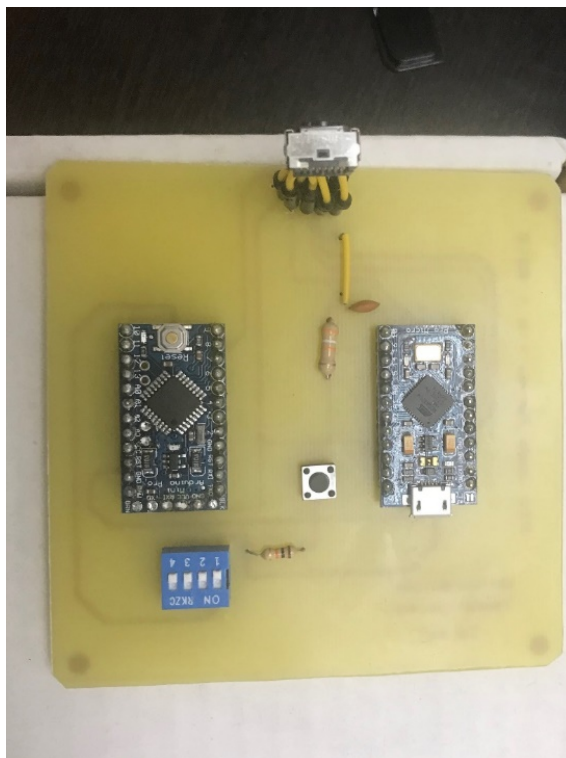


Figura. 3. 16. Placa de circuito impreso vista desde el lado de los componentes

Fuente: Elaborado por el autor

Para el lápiz, que es la interfaz de usuario del sistema, y se requiere que sea fácil de manejar como si fuese un marcador estándar; el camino más sencillo fue emplear el cuerpo de un marcador de pizarra líquida, reemplazando la punta por un diodo LED infrarrojo y usando el cuerpo restante para soportar las baterías. Como se observa en la figura 3.17.



Figura. 3. 17. Punta del marcador, se observa el interruptor pulsador y el LED IR

Fuente: Elaborado por el autor

Este diseño permite que la batería (que es la parte más pesada) descansa sobre la palma de la mano al usar el marcador. La idea es que su uso sea lo más natural posible para que la manipulación no se sienta forzada, como indica la figura 3.18.



Figura. 3. 18. Modo de uso del lápiz IR

Fuente: Elaborado por el autor

Con todos los componentes construidos es necesario realizar pruebas de operación. Dichas pruebas deben permitir validar la operación del prototipo en un ambiente controlado, se debe confirmar que existe información, que la operación básica deseada se cumpla y que, en general, el dispositivo esté operando.

3.1: PRUEBAS

Las pruebas se realizaron cuando todo fue ensamblado. Consistieron en confirmar que el módulo esté enviando señales de posicionamiento y que la señal sea apropiada para el uso en el computador. Esto se logra abriendo el puerto serie en el Arduino IDE (*Ctrl+Shift+M*), el sistema – en ausencia de luz infrarroja – mostrará lo indicado en la figura 3.19.

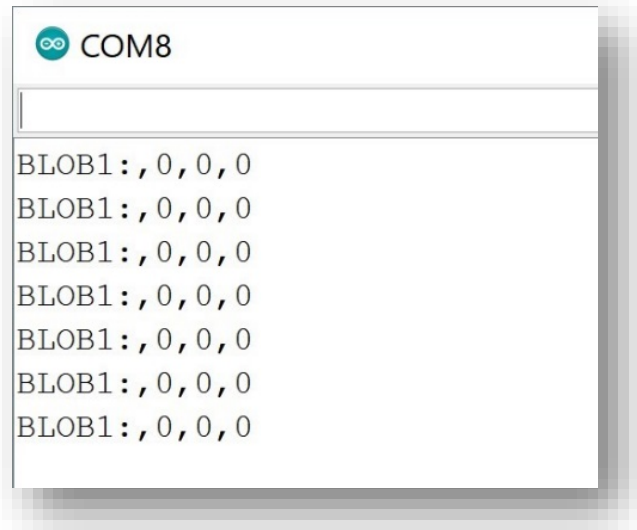


Figura. 3. 19. Ventana del puerto serie en ausencia de luz infrarroja

Fuente: Elaborado por el autor

Si se presenta una fuente infrarroja el sistema debe indicar las coordenadas relativas de la posición, en base a su ubicación, como se ve en la figura 3.20.

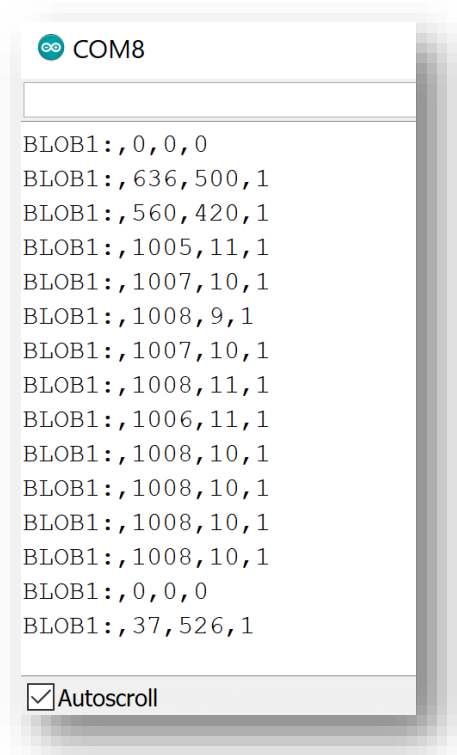


Figura. 3. 20. Ventana de posición con una fuente de luz infrarroja

Fuente: Elaborado por el autor

Con la información presentada en la figura 3.20 se puede comprobar que el módulo está enviando la información y es procesada de manera adecuada.

Otra prueba de operación se realizó encendiendo la funcionalidad de ratón (mouse) y observando el comportamiento en la computadora. Se observa que al mover el lápiz infrarrojo el cursor del ratón de la computadora respondió a los movimientos.

Ahora bien, existe una aplicación que permite también procesar la información enviada por el dispositivo en relación con el tamaño del blob y su posición, esta aplicación se llama Processing y es empleada por desarrolladores de todo el mundo para procesar resultados de aplicaciones diseñadas. En esta ventana se puede observar el comportamiento del dispositivo sin mover el ratón del sistema operativo, como se aprecia en la figura 3.21.

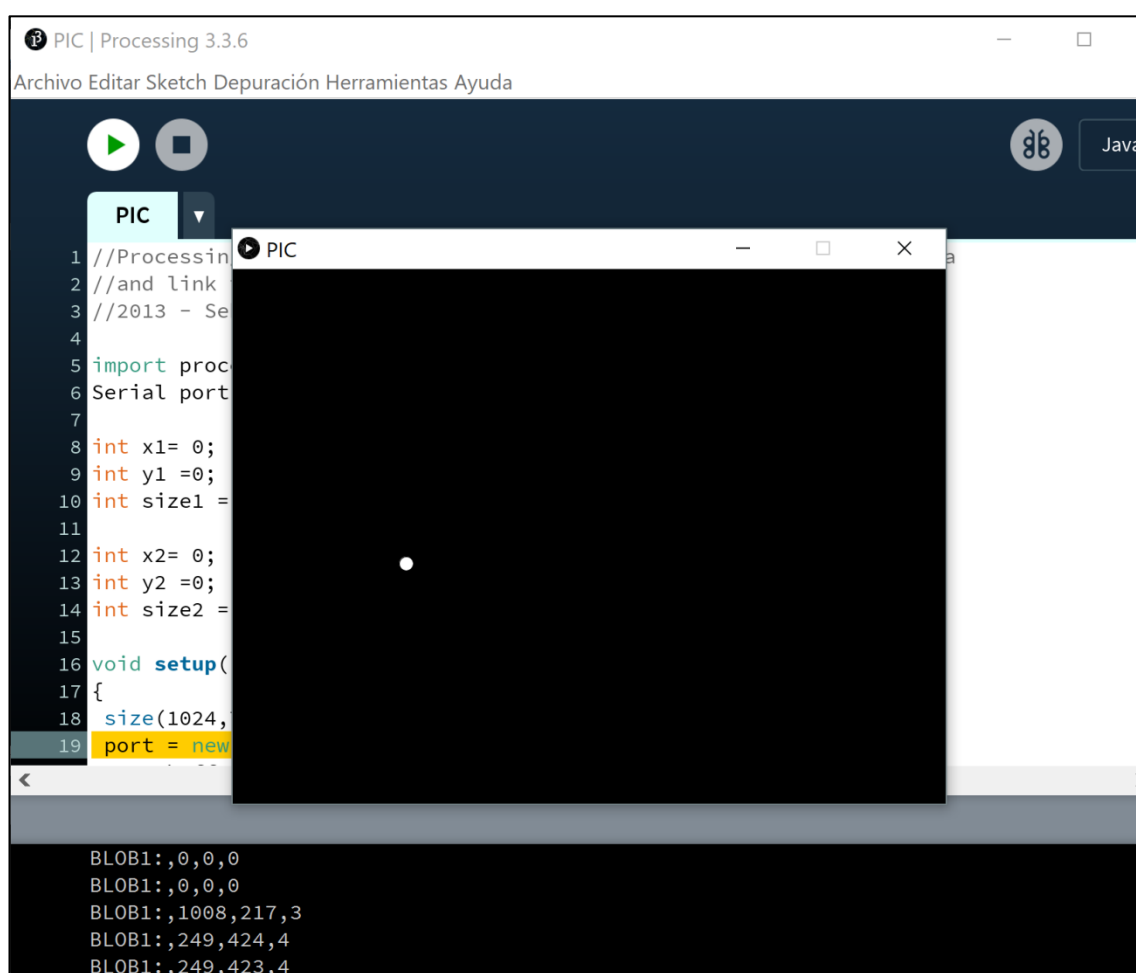


Figura. 3. 21. Ventana de Processing

Indica de manera gráfica la posición del lápiz infrarrojo (mancha blanca en la pantalla), indicando las coordenadas en la parte inferior.

Fuente: Elaborado por el autor

3.2: ANÁLISIS DE RESULTADOS

En base a las dos pruebas realizadas, los resultados se observan en la tabla 3.5.:

Tabla. 3. 5. Tabla comparativa de resultado

Prueba	Resultado
En ausencia de luz infrarroja el dispositivo emite información nula de coordenadas	El dispositivo, en ausencia de luz infrarroja, emite coordenadas 0,0
En presencia de luz infrarroja el dispositivo emite información de coordenadas	El dispositivo, en presencia de luz infrarroja, emite coordenadas relativas
En presencia de luz el dispositivo es sensible	El dispositivo presenta una operación extremadamente sensible, el menor movimiento de la luz hace que el resultado de la operación se mueva demasiado rápido
Observar el comportamiento del dispositivo en estado de reposo	Se nota que en reposo (sin fuente de luz) a momentos el dispositivo presenta <i>blobs</i> que son causados como ruido por la luz natural
Activar la operación de “modo ratón” y observar comportamiento	Con el “modo ratón” activado el movimiento de la fuente de luz hace que el cursor del ratón del sistema operativo se mueva en concordancia con el movimiento de la fuente de luz
Con “modo ratón” activo el cursor debe responder al movimiento de la fuente de luz	Se observa que el sistema responde, aunque de manera muy sensible.

Fuente: Elaborado por el autor

Prueba 1: Si no existe ninguna luz infrarroja el dispositivo envía coordenadas 0,0; indicando ausencia del lápiz infrarrojo. Al presentar el lápiz el dispositivo indica su posición relativa dentro del campo de visión de este enviando coordenadas, se observó que el comportamiento del dispositivo es extremadamente sensible, por lo que fue necesario aumentar un retardo en el programa de manera de poder observar de mejor manera su operación.

A momentos el sistema indica coordenadas y *blobs* sin presencia del lápiz infrarrojo, lo cual es una interferencia no deseada, esto sucede durante el día en sitios con ventanas o cierto tipo de iluminación, la solución para esto es instalar un “filtro infrarrojo” frente al módulo. Este filtro no es más que un plástico de color violeta, similar al usado en muchos controles remotos, que actúa como filtro a ciertas luces, eliminando por completo la interferencia.

Prueba 2: El sistema es capaz de controlar al ratón del sistema operativo, aunque a momentos es muy sensible, un movimiento leve del lápiz infrarrojo mueve rápidamente al ratón del sistema operativo, esto se corrigió aumentando un retraso en el programa de procesamiento de la señal.

CONCLUSIONES

- El resultado de este proyecto demuestra que es posible crear dispositivos electrónicos de bajo costo para solucionar requerimientos del mundo real, fuera de pruebas de laboratorio e incluso con capacidad de hacer un desarrollo comercial. El dispositivo cumple con la función de brindar interacción con un proyector, brindando además otras posibles funciones (para futuros desarrollos) como: interacción con PC “*touch*”, capacidad de implementar múltiples ratones al tiempo en el sistema operativo entre otras. Con esto se ha cumplido el objetivo general del proyecto.
- El uso de componentes de electrónica digital (dispositivos programables) permitió que la creación del dispositivo sea cumplida completamente, permitiendo mejoras futuras y desarrollo de la idea. Esto cumple con el objetivo de definir los elementos constitutivos de un dispositivo electrónico para transformar un proyector estándar en interactivo.
- Debido al uso de dispositivos programables el uso de buenas técnicas de programación y diseño (diagrama de bloques, diagrama de flujo, y similares) permitió el desarrollo de la idea de manera práctica, con un desarrollo funcional casi inmediato. Cumpliendo así el objetivo de elaboración del diagrama de bloques.
- Partiendo de los pasos ordenados indicados arriba se facilitó el desarrollo del diagrama esquemático del circuito. Realizando un análisis del flujo y tratamiento de la señal el proceso de diseño del circuito electrónico se transforma en un proceso intuitivo, permitiendo así, cumplir con el objetivo trazado.
- De igual manera, una vez planificado diagrama de bloques, flujo, esquemático y definido el procesador a usar la creación o modificación del algoritmo de programación se transformó en una tarea no compleja.
- Con todos los pasos cumplidos (diseño del circuito electrónico y algoritmo de programación) se procedió a implementar el dispositivo.
- El definir un esquema de pruebas permitió que se valide la operación deseada del dispositivo, facilita el proceso de corrección de errores y permite una mejora continua del mismo (de ser deseado) para futuros desarrollos.

RECOMENDACIONES

En el mercado existen varios tipos de microcontroladores; para una implementación sencilla de tipo experimental, como la planteada en el presente proyecto, el uso de dispositivos que puedan ser programados sin necesidad de removerlos de la placa de circuito impreso facilita mucho el trabajo, pues remover el microcontrolador de la placa electrónica en las pruebas puede generar daños además de generar retrasos, gastos no planificados y cuestiones similares.

En este proyecto se emplea un protocolo de comunicación propietario entre los circuitos integrados (I²C), el cual tiene ciertas características que complican un poco el diseño; por ejemplo, el hecho de que la línea de datos sea de 3,3 voltios limita bastante la elección del microcontrolador, pues muchos de los dispositivos disponibles comercialmente emplean una línea de datos de 5 voltios. Esto puede causar que sean requeridos dispositivos adicionales para el ajuste de señal, lo cual complica el diseño, aumenta los costos y disminuye la confiabilidad.

Como todo dispositivo electrónico se debe validar su operación mediante pruebas de funcionamiento; con las cuales, si el dispositivo no funciona como se espera, se puede corregir errores para lograr la funcionalidad deseada.

Previo a iniciar el proceso de diseño, cuando se están analizando opciones de dispositivos a emplear, se debe analizar a detalle las funciones y características de los dispositivos elegidos; pues al ser programables, el diseñador puede encontrar situaciones en las cuales éstos no funcionen de la manera esperada, tengan falta de funciones o no sean adecuados para el uso planificado. Esto generará frustración en el diseñador, aumento de costos, plazos o que simplemente el prototipo no sea viable.

BIBLIOGRAFÍA

- Arduino. (2018). *Arduino - Home*. Obtenido de <https://www.arduino.cc/en/Reference/>
- Derwyn. (04 de Junio de 2017). *Arduino NANO as ISP (programmer)*. Obtenido de <https://gw4sae.wordpress.com/2017/06/04/arduino-nano-as-isp-programmer/>
- Hämmerling, M. (01 de June de 2008). *Engbedded AVR Fuse Calculator*. Obtenido de <http://www.engbedded.com/fusecalc>
- Hobley, S. (22 de Febrero de 2009). *www.stephenhobley.com*. Obtenido de www.stephenhobley.com: <http://www.stephenhobley.com>
- Lee, J. C. (2 de Noviembre de 2016). *Procastineering - Project blog*. Obtenido de <http://procastineering.blogspot.com/>
- Microsoft. (09 de Septiembre de 2011). *Interpreting Mouse Axis Data*. Obtenido de [https://docs.microsoft.com/en-us/previous-versions/windows/desktop/ee418272\(v=vs.85\)](https://docs.microsoft.com/en-us/previous-versions/windows/desktop/ee418272(v=vs.85))
- Simple Machines. (2017). *Wii mote Project*. Obtenido de <https://www.wiimoteproject.com/index.php>
- Sparkfun Electronics. (13 de Marzo de 2016). *Using the Arduino Pro Mini*. Obtenido de Learn Sparkfun: <https://learn.sparkfun.com/tutorials/using-the-arduino-pro-mini-33v>
- Wikipedia. (2006). *Wii Remote*. Obtenido de https://en.wikipedia.org/wiki/Wii_Remote

ANEXO 1

MANUAL TÉCNICO DEL DISPOSITIVO

Dispositivo electrónico para transformar un proyector estándar en interactivo

El dispositivo se compone de dos partes, una parte fija que contiene los dispositivos electrónicos y un lápiz infrarrojo que es la parte móvil. Para la conexión al equipo de interacción se debe usar un cable USB estándar con conector tipo A (macho) en un extremo y micro-A (micro-USB) en el otro extremo. Los componentes generales se observan en la figura 1.

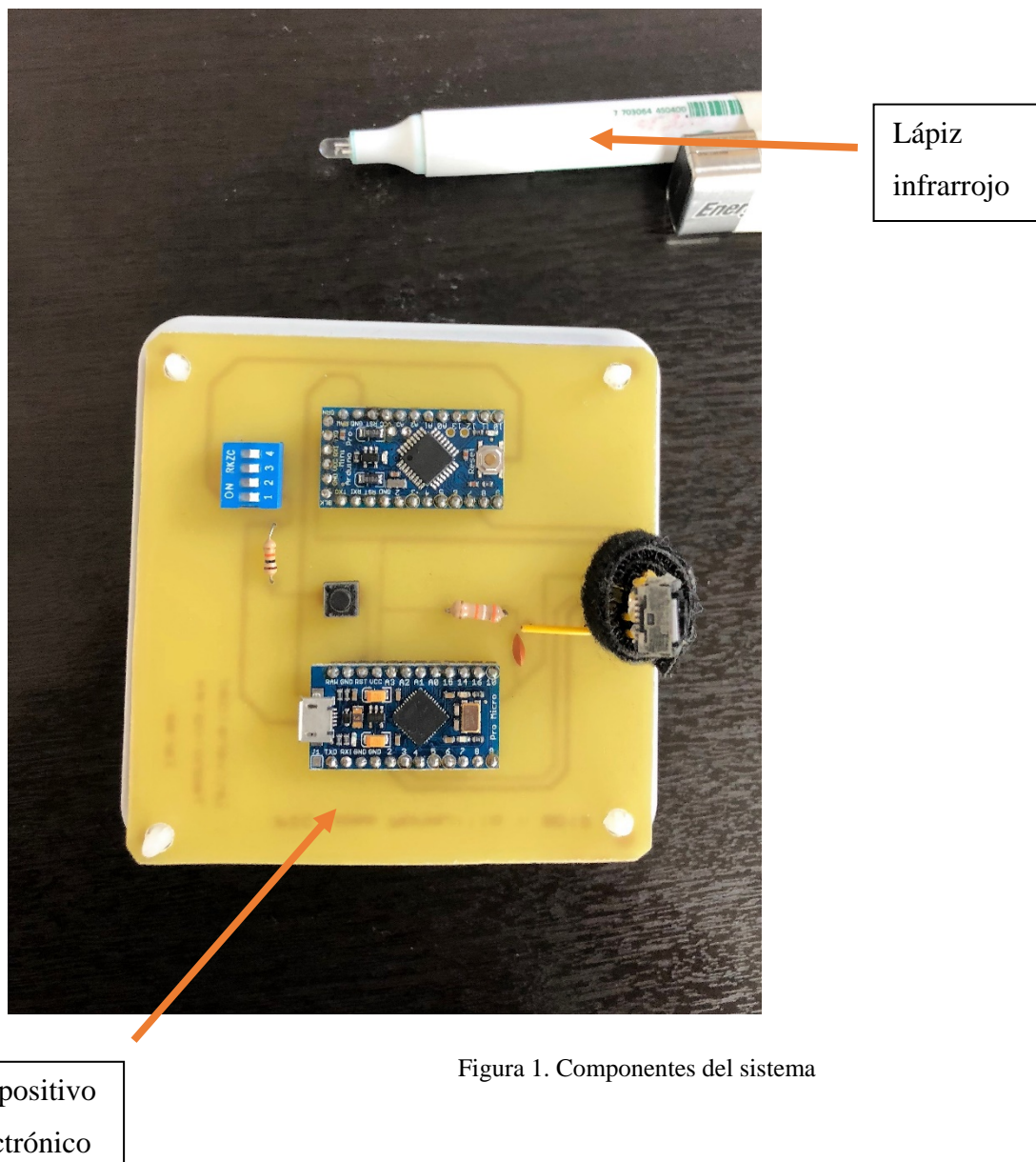


Figura 1. Componentes del sistema

Parte 1. Instalación

El dispositivo debe estar instalado en un ambiente seco, a temperatura ambiente y debe ser apuntado hacia la superficie donde se proyectará la imagen. En dicha superficie será donde se use el lápiz interactivo. La parte que debe observar hacia la pantalla es el sensor de posicionamiento, como se observa en la figura 2.

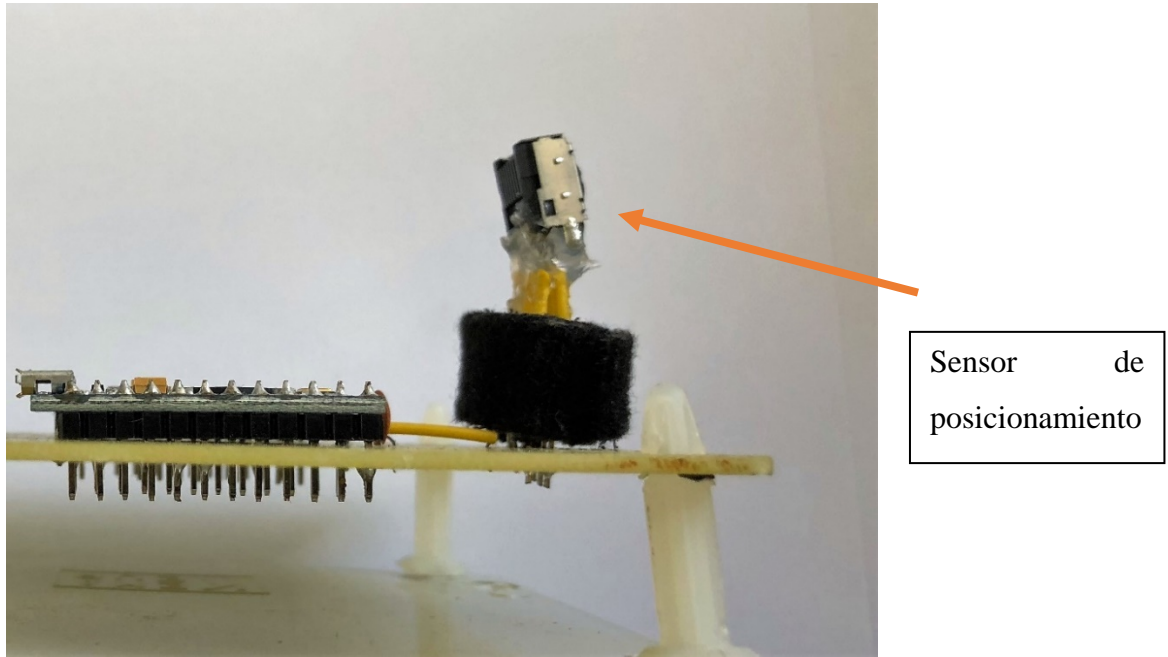


Figura 2. El sensor o módulo de posicionamiento

El dispositivo debe estar instalado en una superficie plana, con el sensor orientado hacia arriba.

Debido a la “apertura” del lente el dispositivo se debe instalar de manera que “vea” toda la superficie de la proyección, en la tabla 1 se muestran las distancias recomendadas de uso. No se recomienda una distancia mayor a 2,5 metros.

Tabla 1. Distancias de uso

Distancia entre pantalla y dispositivo (m)	Ancho de pantalla (cm)	Alto de pantalla (cm)
1	100	85
1,5	150	130
2	200	170

2,5	250	210
-----	-----	-----

Parte 2. Conexión

Una vez que el dispositivo está instalado se debe conectar el cable USB, el terminal micro-A (figura 3a) se inserta con cuidado en el terminal del módulo electrónico (figura 4). El otro terminal del cable tipo A (figura 3b) se inserta en el puerto del computador (figura 5).



Figura 3a. Terminal micro-A (micro-USB)

Figura 3b. Terminal tipo A (USB)

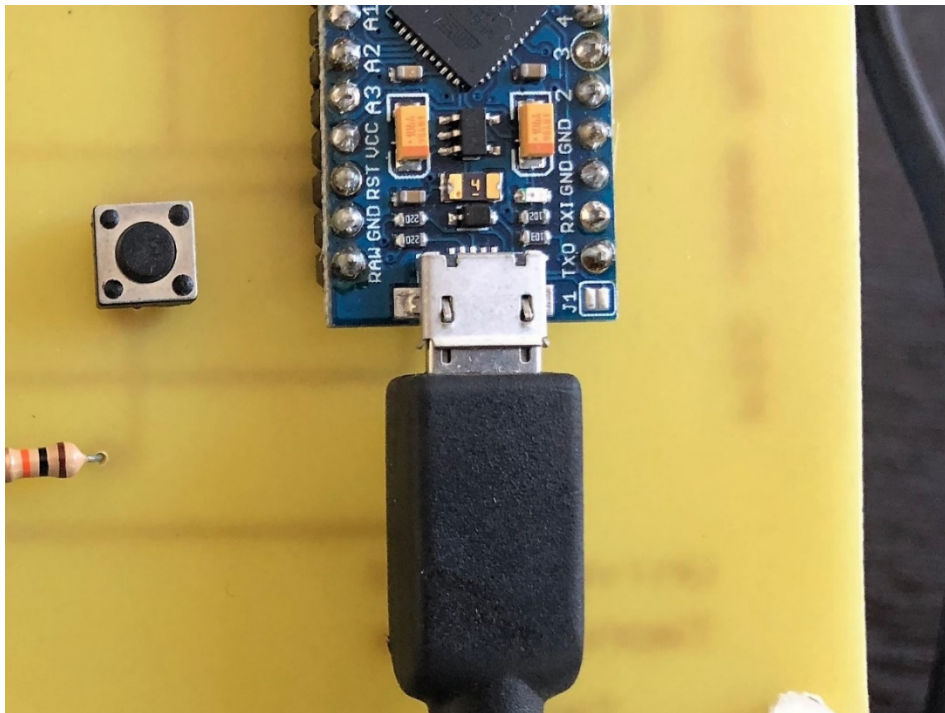


Figura 4. Conexión del cable micro-A en el dispositivo



Figura 5. Conexión del cable USB tipo A en el computador

Al conectar todo algunos iluminadores LED encenderán en el dispositivo, indicando que está alimentado, esto se observa en la figura 6.



Figura 6. Estado encendido del dispositivo

Parte 3. Uso del dispositivo

Para usar el equipo (activar el modo ratón) se debe mover el interruptor 1 a la posición de ON, como se observa en la figura 7.

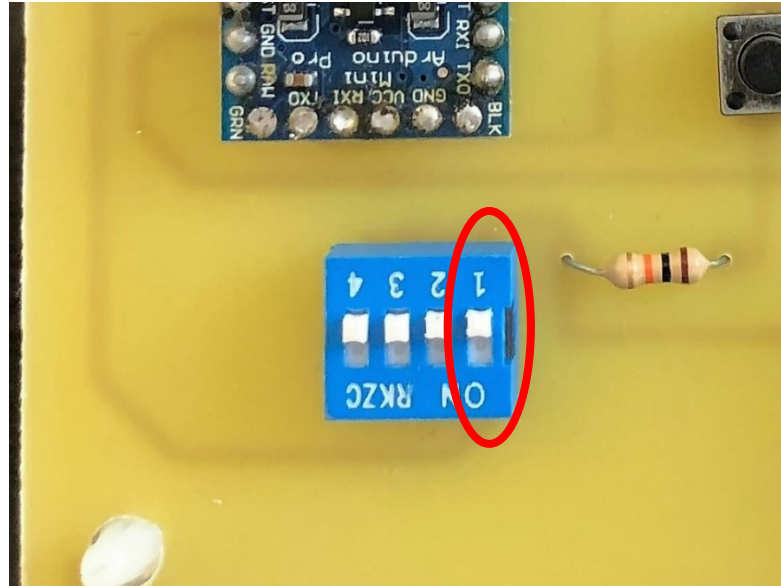


Figura 7. Encendido del modo ratón

Con esto el dispositivo queda listo para operación, al realizar la proyección hay que tener cuidado de no posicionarse frente al lápiz infrarrojo o bloquear la imagen de la módulo. Durante el primer uso el sistema operativo del computador instalará controladores del ratón, antes del uso se debe esperar este proceso, normalmente no demora más de 30 segundos.

Para activar la interactividad se debe presionar el botón del lápiz para que éste emita señal infrarroja, para detener la interactividad simplemente se debe dejar de presionar el botón, tal como se observa en la Figura 8, y la operación en la Figura 9.



Figura 8. Botón de encendido del lápiz infrarrojo



Figura 9. Interacción con una imagen proyectada

Para apagar el dispositivo, primero se debe mover el interruptor de la placa electrónica a la posición OFF, y luego desconectar el cable USB.

El dispositivo no requiere limpieza ni calibración, no hay partes de “servicio de usuario”, al almacenar se debe tener cuidado de no forzar ninguna conexión.

ANEXO 2. ALGORITMO DEL PROGRAMA

```
/*  
Sketch para control de raton mediante lectura de posicion de camara IR  
Jose Ronquillo 2018  
Universidad Tecnologica Israel  
*/  
  
#include <Wire.h>           //Libreria de comunicacion I2C  
#include <PVision.h>       //Libreria de comunicacion camara  
#include <Mouse.h>         //Libreria de raton USB  
  
PVision ircam;             //detalle la camara  
byte result;               //el resultado de la informacion que envia la camara es un byte  
  
//inicio el pin del switch que inicia el raton y los ejes  
const int switchPin = 9;   //el switch que prendera y apagara el modo raton  
const int xAxis = {ircam.Blob1.X}; //declaro al eje x(0)  
const int yAxis = {ircam.Blob1.Y}; //declaro al eje y(1)  
  
//pongo los parametros de lectura  
int range = 12;           //pongo el rango de movimiento de X y Y  
int responseDelay = 2;    //pongo un pequeño retraso al movimiento en ms  
int threshold = range / 4; //parametros de descanso  
int center = range / 2;   //declaro el centro o descanso  
  
boolean mouseIsActive = false; //indico si el raton sera o no controlado  
int lastSwitchState = LOW;     //estado previo del switch  
  
void setup()  
{  
  Serial.begin(9600);         //activo el puerto serie en 9600 baudios  
  ircam.init();              //inicializo la camara  
  
  pinMode (switchPin, INPUT); //declaro el pin como entrada  
  Mouse.begin ();  
}  
  
void loop()  
{  
  int switchState = digitalRead(switchPin); //aqui leo el estado del switch
```

```
if (switchState != lastSwitchState){           //si ha cambiado y esta alto, cambiamos el estado del raton
  if (switchState == HIGH) {
    mouseIsActive = !mouseIsActive;
  }
}

lastSwitchState = switchState;                 //guardo el estado del switch para la proxima interaccion

result = ircam.read();                         //leo la camara IR

if (result & BLOB1)                            //si detecta un IR Blob
{
  Serial.print("BLOB1:");                       //indicamos Blob1
  Serial.print(ircam.Blob1.X);                  //salida eje X del Blob
  Serial.print(',');
  Serial.print(ircam.Blob1.Y);                 //salida eje Y del Blob
  Serial.print(',');
  Serial.print(ircam.Blob1.Size);              //tamaño del Blob
  int xReading = ircam.Blob1.X;                //mando el dato de X a la lectura del raton
  int yReading = ircam.Blob1.Y;                //mando el dato de Y a la lectura del raton

  //si el control de raton esta activo
  if (mouseIsActive){
    Mouse.move (xReading, yReading, 0);        //muevo las lecturas de X y Y al mov del raton
  }

  //si el size es mayor que 0 presiono y mantengo el boton izquierdo del raton
  if ((ircam.Blob1.Size) > 0) {
    if (!Mouse.isPressed (MOUSE_LEFT)){
      Mouse.press (MOUSE_LEFT);
    }
  }
  else {
    if (Mouse.isPressed(MOUSE_LEFT)){
      Mouse.release(MOUSE_LEFT);
    }
  }
}

delay(responseDelay);

// Aqui escalo el rango de lectura de X y escalo el rango de entrada entre 0 y <rango>

int readAxis(xAxis); {
```

```
int reading = ircam.Blob1.X; //leo el eje
reading = map(reading, 0, 1023, 0, range); //mapeo la lectura entre los rangos

int distance = reading - center;
if (abs(distance) < threshold) {
    distance = 0;
}

return distance;
}

// Aqui escalo el rango de lectura de Y y escalo el rango de entrada entre 0 y <rango>

int readAxisy (yAxis); {
int reading = ircam.Blob1.Y; //leo el eje
reading = map(reading, 0, 1023, 0, range); //mapeo la lectura entre los rangos
int distance = reading - center;
if (abs(distance) < threshold) {
    distance = 0;
}
return distance;
}

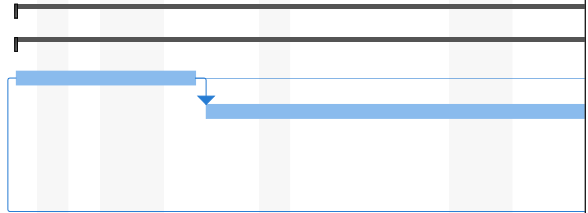
{
if (!(result & BLOB1)) //si no hay Blob, mandamos 0 en X y Y
{
Serial.print("BLOB1:");
Serial.print("0");
Serial.print(',');
Serial.print("0");
Serial.print(',');
Serial.print("0");
}
}

Serial.println("");

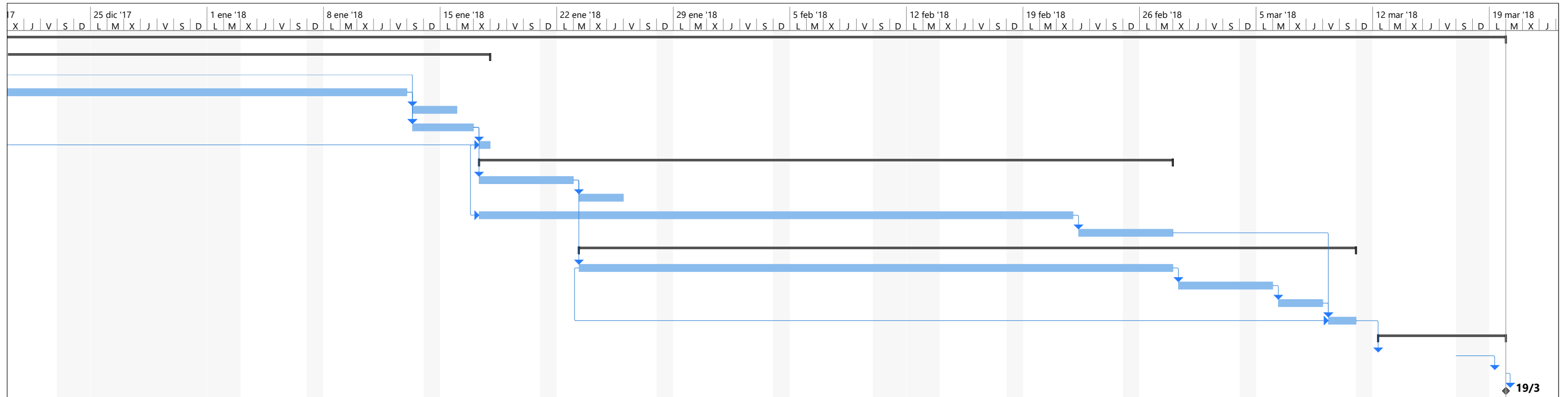
// Breve retardo...
delay(100);
```

ANEXO 3. CRONOGRAMA DE DESARROLLO DEL PROYECTO

Id	Modo de tarea	Nombre de tarea	Duración	Comienzo	Fin	Predecesoras	17	30 oct '17	6 nov '17	13 nov '17	20 nov '17	27 nov '17	4 dic '17	11 dic '17	18 dic '17																																													
							X	J	V	S	D	L	M	X	J	V	S	D	L	M	X	J	V	S	D	L	M	X	J	V	S	D	L	M	X	J	V	S	D	L	M	X	J	V	S	D														
1		PIC Jose Ronquillo	76 días?	sáb 2/12/17	lun 19/3/18																																																							
2		Investigación & Escrito	28 días?	sáb 2/12/17	mié 17/1/18																																																							
3		Diagramado de plan de investigac	3 días	sáb 2/12/17	jue 7/12/17																																																							
4		Desarrollo de investigacion	21 días	vie 8/12/17	vie 12/1/18	3																																																						
5		Análisis de resultados	2 días	sáb 13/1/18	lun 15/1/18	4																																																						
6		Recopilación de lo investigado	3 días	sáb 13/1/18	mar 16/1/18	3;4																																																						
7		Inicio del desarrollo del trabajo esc	1 día?	mié 17/1/18	mié 17/1/18	3CC;6																																																						
8		Diseño	33 días	mié 17/1/18	mar 27/2/18																																																							
9		Prediseño, simulación	5 días	mié 17/1/18	lun 22/1/18	6																																																						
10		Prediseño, documentacion disposit	3 días	mar 23/1/18	jue 25/1/18	9																																																						
11		Desarrollo del trabajo escrito	28 días	mié 17/1/18	mié 21/2/18	7CC																																																						
12		Entrega a revisión del trabajo escrit	5 días	jue 22/2/18	mar 27/2/18	11																																																						
13		Construcción	38 días	mar 23/1/18	sáb 10/3/18																																																							
14		Construcción del prototipo	28 días	mar 23/1/18	mar 27/2/18	9																																																						
15		Prueba y correccion de errores	5 días	mié 28/2/18	lun 5/3/18	14																																																						
16		Implementación del prototipo	3 días	mar 6/3/18	jue 8/3/18	15																																																						
17		Documentación del prototipo	2 días	vie 9/3/18	sáb 10/3/18	12;14CC;16																																																						
18		Defensa y entrega	6 días	lun 12/3/18	lun 19/3/18																																																							
19		Cambios en el trabajo escrito	5 días	lun 12/3/18	vie 16/3/18	17																																																						
20		Entrega del trabajo escrito	1 día	lun 19/3/18	lun 19/3/18	19																																																						
21		Fin	0 días	lun 19/3/18	lun 19/3/18	20																																																						



Proyecto: PIC Jose Ronquillo Fecha: mié 5/9/18	Tarea		Resumen del proyecto		Tarea manual		solo el comienzo		Fecha límite	
	División		Tarea inactiva		solo duración		solo fin		Progreso	
	Hito		Hito inactivo		Informe de resumen manual		Tareas externas		Progreso manual	
	Resumen		Resumen inactivo		Resumen manual		Hito externo			



Proyecto: PIC Jose Ronquillo
 Fecha: mié 5/9/18

Tarea	Resumen del proyecto	Resumen inactivo	Tarea manual	solo el comienzo	Fecha límite
División	Tarea inactiva	Hito inactivo	solo duración	solo fin	Progreso
Hito	Hito inactivo	Resumen inactivo	Informe de resumen manual	Tareas externas	Progreso manual
Resumen	Resumen inactivo	Resumen manual	Hito externo	Hito externo	

ANEXO 4: HOJA DE ESPECIFICACIONES TÉCNICAS PROCESADOR ATMEGA328

Features

- High Performance, Low Power Atmel® AVR® 8-Bit Microcontroller
- Advanced RISC Architecture
 - 131 Powerful Instructions – Most Single Clock Cycle Execution
 - 32 x 8 General Purpose Working Registers
 - Fully Static Operation
 - Up to 20 MIPS Throughput at 20MHz
 - On-chip 2-cycle Multiplier
- High Endurance Non-volatile Memory Segments
 - 4/8/16/32KBytes of In-System Self-Programmable Flash program memory
 - 256/512/512/1KBytes EEPROM
 - 512/1K/1K/2KBytes Internal SRAM
 - Write/Erase Cycles: 10,000 Flash/100,000 EEPROM
 - Data retention: 20 years at 85°C/100 years at 25°C⁽¹⁾
 - Optional Boot Code Section with Independent Lock Bits
In-System Programming by On-chip Boot Program
True Read-While-Write Operation
 - Programming Lock for Software Security
- Atmel® QTouch® library support
 - Capacitive touch buttons, sliders and wheels
 - QTouch and QMatrix® acquisition
 - Up to 64 sense channels
- Peripheral Features
 - Two 8-bit Timer/Counters with Separate Prescaler and Compare Mode
 - One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode
 - Real Time Counter with Separate Oscillator
 - Six PWM Channels
 - 8-channel 10-bit ADC in TQFP and QFN/MLF package
Temperature Measurement
 - 6-channel 10-bit ADC in PDIP Package
Temperature Measurement
 - Programmable Serial USART
 - Master/Slave SPI Serial Interface
 - Byte-oriented 2-wire Serial Interface (Philips I²C compatible)
 - Programmable Watchdog Timer with Separate On-chip Oscillator
 - On-chip Analog Comparator
 - Interrupt and Wake-up on Pin Change
- Special Microcontroller Features
 - Power-on Reset and Programmable Brown-out Detection
 - Internal Calibrated Oscillator
 - External and Internal Interrupt Sources
 - Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby, and Extended Standby
- I/O and Packages
 - 23 Programmable I/O Lines
 - 28-pin PDIP, 32-lead TQFP, 28-pad QFN/MLF and 32-pad QFN/MLF
- Operating Voltage:
 - 1.8 - 5.5V
- Temperature Range:
 - -40°C to 85°C
- Speed Grade:
 - 0 - 4MHz@1.8 - 5.5V, 0 - 10MHz@2.7 - 5.5.V, 0 - 20MHz @ 4.5 - 5.5V
- Power Consumption at 1MHz, 1.8V, 25°C
 - Active Mode: 0.2mA
 - Power-down Mode: 0.1µA
 - Power-save Mode: 0.75µA (Including 32kHz RTC)



8-bit Atmel Microcontroller with 4/8/16/32K Bytes In-System Programmable Flash

ATmega48A
ATmega48PA
ATmega88A
ATmega88PA
ATmega168A
ATmega168PA
ATmega328
ATmega328P

Summary

Rev. 8271DS-AVR-05/11



1. Pin Configurations

Figure 1-1. Pinout ATmega48A/PA/88A/PA/168A/PA/328/P

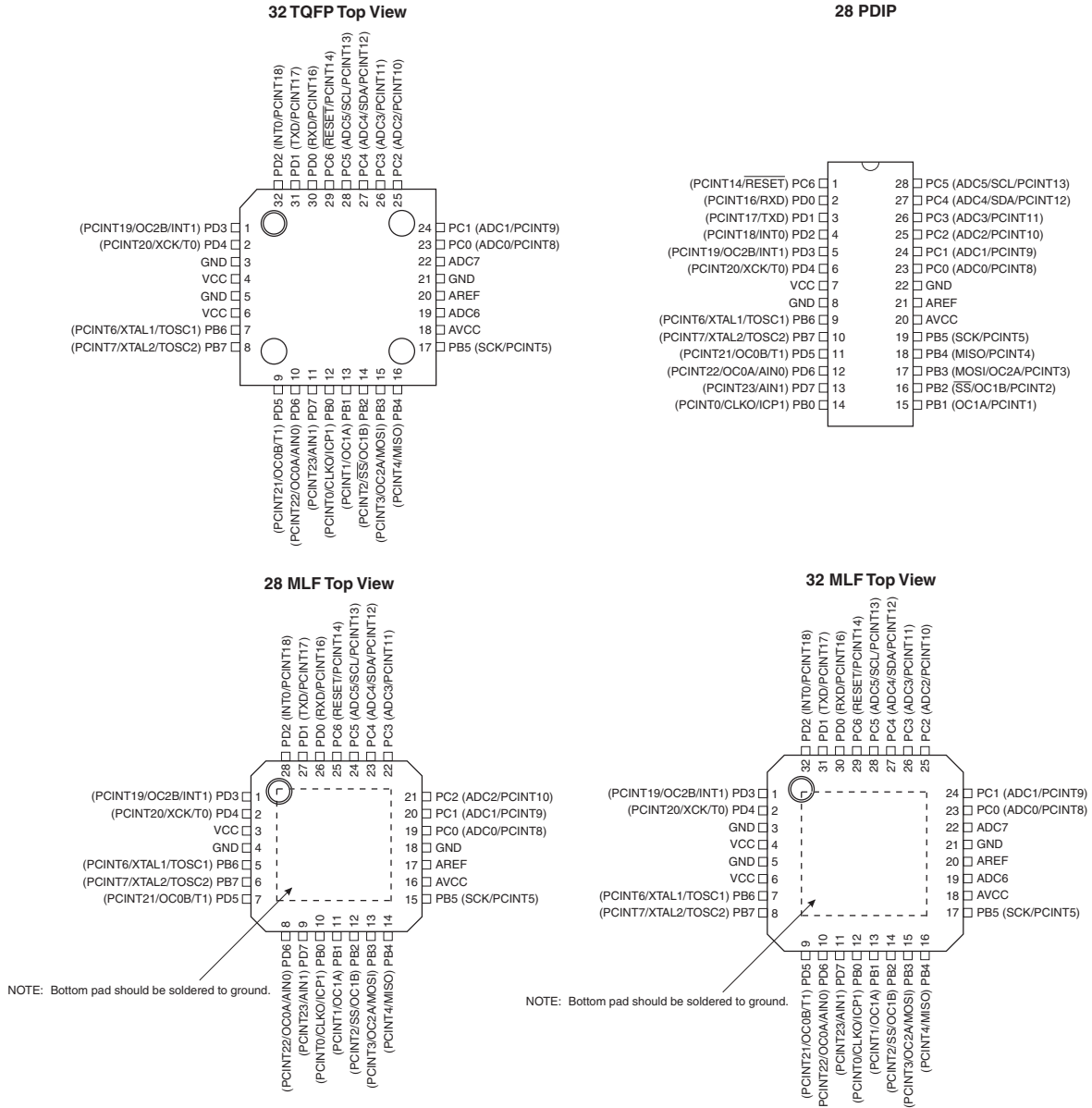


Table 1-1. 32UFBGA - Pinout ATmega48A/48PA/88A/88PA/168A/168PA

	1	2	3	4	5	6
A	PD2	PD1	PC6	PC4	PC2	PC1
B	PD3	PD4	PD0	PC5	PC3	PC0
C	GND	GND			ADC7	GND
D	VDD	VDD			AREF	ADC6
E	PB6	PD6	PB0	PB2	AVDD	PB5
F	PB7	PD5	PD7	PB1	PB3	PB4

1.1 Pin Descriptions

1.1.1 VCC

Digital supply voltage.

1.1.2 GND

Ground.

1.1.3 Port B (PB7:0) XTAL1/XTAL2/TOSC1/TOSC2

Port B is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port B output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port B pins that are externally pulled low will source current if the pull-up resistors are activated. The Port B pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Depending on the clock selection fuse settings, PB6 can be used as input to the inverting Oscillator amplifier and input to the internal clock operating circuit.

Depending on the clock selection fuse settings, PB7 can be used as output from the inverting Oscillator amplifier.

If the Internal Calibrated RC Oscillator is used as chip clock source, PB7...6 is used as TOSC2...1 input for the Asynchronous Timer/Counter2 if the AS2 bit in ASSR is set.

The various special features of Port B are elaborated in ["Alternate Functions of Port B" on page 84](#) and ["System Clock and Clock Options" on page 27](#).

1.1.4 Port C (PC5:0)

Port C is a 7-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The PC5...0 output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port C pins that are externally pulled low will source current if the pull-up resistors are activated. The Port C pins are tri-stated when a reset condition becomes active, even if the clock is not running.

1.1.5 PC6/ $\overline{\text{RESET}}$

If the RSTDISBL Fuse is programmed, PC6 is used as an I/O pin. Note that the electrical characteristics of PC6 differ from those of the other pins of Port C.

If the RSTDISBL Fuse is unprogrammed, PC6 is used as a Reset input. A low level on this pin for longer than the minimum pulse length will generate a Reset, even if the clock is not running. The minimum pulse length is given in [Table 29-12 on page 324](#). Shorter pulses are not guaranteed to generate a Reset.

The various special features of Port C are elaborated in ["Alternate Functions of Port C" on page 87](#).

1.1.6 Port D (PD7:0)

Port D is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port D output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port D pins that are externally pulled low will source current if the pull-up resistors are activated. The Port D pins are tri-stated when a reset condition becomes active, even if the clock is not running.

The various special features of Port D are elaborated in ["Alternate Functions of Port D"](#) on page 90.

1.1.7 **AV_{CC}**

AV_{CC} is the supply voltage pin for the A/D Converter, PC3:0, and ADC7:6. It should be externally connected to V_{CC}, even if the ADC is not used. If the ADC is used, it should be connected to V_{CC} through a low-pass filter. Note that PC6...4 use digital supply voltage, V_{CC}.

1.1.8 **AREF**

AREF is the analog reference pin for the A/D Converter.

1.1.9 **ADC7:6 (TQFP and QFN/MLF Package Only)**

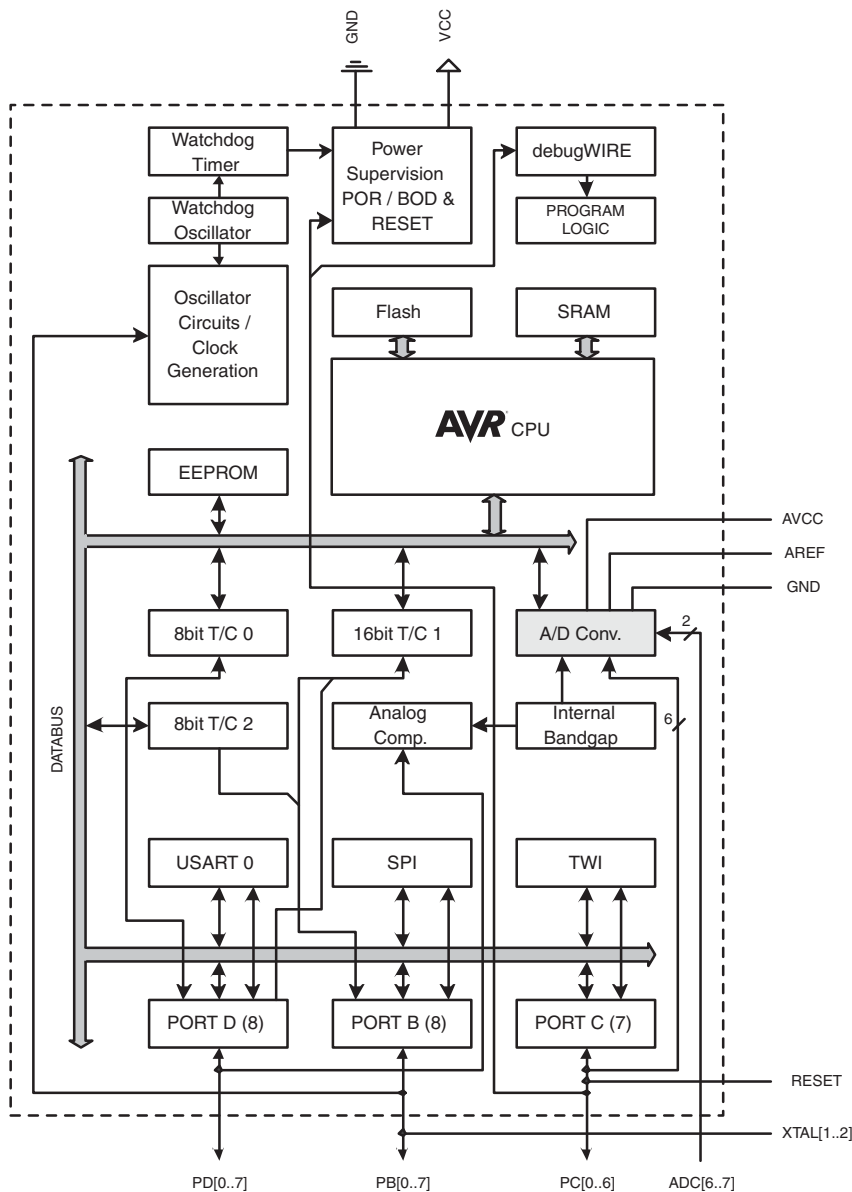
In the TQFP and QFN/MLF package, ADC7:6 serve as analog inputs to the A/D converter. These pins are powered from the analog supply and serve as 10-bit ADC channels.

2. Overview

The ATmega48A/PA/88A/PA/168A/PA/328/P is a low-power CMOS 8-bit microcontroller based on the AVR enhanced RISC architecture. By executing powerful instructions in a single clock cycle, the ATmega48A/PA/88A/PA/168A/PA/328/P achieves throughputs approaching 1 MIPS per MHz allowing the system designer to optimize power consumption versus processing speed.

2.1 Block Diagram

Figure 2-1. Block Diagram



The AVR core combines a rich instruction set with 32 general purpose working registers. All the 32 registers are directly connected to the Arithmetic Logic Unit (ALU), allowing two independent registers to be accessed in one single instruction executed in one clock cycle. The resulting

architecture is more code efficient while achieving throughputs up to ten times faster than conventional CISC microcontrollers.

The ATmega48A/PA/88A/PA/168A/PA/328/P provides the following features: 4K/8Kbytes of In-System Programmable Flash with Read-While-Write capabilities, 256/512/512/1Kbytes EEPROM, 512/1K/1K/2Kbytes SRAM, 23 general purpose I/O lines, 32 general purpose working registers, three flexible Timer/Counters with compare modes, internal and external interrupts, a serial programmable USART, a byte-oriented 2-wire Serial Interface, an SPI serial port, a 6-channel 10-bit ADC (8 channels in TQFP and QFN/MLF packages), a programmable Watchdog Timer with internal Oscillator, and five software selectable power saving modes. The Idle mode stops the CPU while allowing the SRAM, Timer/Counters, USART, 2-wire Serial Interface, SPI port, and interrupt system to continue functioning. The Power-down mode saves the register contents but freezes the Oscillator, disabling all other chip functions until the next interrupt or hardware reset. In Power-save mode, the asynchronous timer continues to run, allowing the user to maintain a timer base while the rest of the device is sleeping. The ADC Noise Reduction mode stops the CPU and all I/O modules except asynchronous timer and ADC, to minimize switching noise during ADC conversions. In Standby mode, the crystal/resonator Oscillator is running while the rest of the device is sleeping. This allows very fast start-up combined with low power consumption.

Atmel® offers the QTouch® library for embedding capacitive touch buttons, sliders and wheels functionality into AVR® microcontrollers. The patented charge-transfer signal acquisition offers robust sensing and includes fully debounced reporting of touch keys and includes Adjacent Key Suppression® (AKS™) technology for unambiguous detection of key events. The easy-to-use QTouch Suite toolchain allows you to explore, develop and debug your own touch applications.

The device is manufactured using Atmel's high density non-volatile memory technology. The On-chip ISP Flash allows the program memory to be reprogrammed In-System through an SPI serial interface, by a conventional non-volatile memory programmer, or by an On-chip Boot program running on the AVR core. The Boot program can use any interface to download the application program in the Application Flash memory. Software in the Boot Flash section will continue to run while the Application Flash section is updated, providing true Read-While-Write operation. By combining an 8-bit RISC CPU with In-System Self-Programmable Flash on a monolithic chip, the Atmel ATmega48A/PA/88A/PA/168A/PA/328/P is a powerful microcontroller that provides a highly flexible and cost effective solution to many embedded control applications.

The ATmega48A/PA/88A/PA/168A/PA/328/P AVR is supported with a full suite of program and system development tools including: C Compilers, Macro Assemblers, Program Debugger/Simulators, In-Circuit Emulators, and Evaluation kits.

2.2 Comparison Between Processors

The ATmega48A/PA/88A/PA/168A/PA/328/P differ only in memory sizes, boot loader support, and interrupt vector sizes. [Table 2-1](#) summarizes the different memory and interrupt vector sizes for the devices.

Table 2-1. Memory Size Summary

Device	Flash	EEPROM	RAM	Interrupt Vector Size
ATmega48A	4KBytes	256Bytes	512Bytes	1 instruction word/vector
ATmega48PA	4KBytes	256Bytes	512Bytes	1 instruction word/vector
ATmega88A	8KBytes	512Bytes	1KBytes	1 instruction word/vector

Table 2-1. Memory Size Summary (Continued)

Device	Flash	EEPROM	RAM	Interrupt Vector Size
ATmega88PA	8KBytes	512Bytes	1KBytes	1 instruction word/vector
ATmega168A	16KBytes	512Bytes	1KBytes	2 instruction words/vector
ATmega168PA	16KBytes	512Bytes	1KBytes	2 instruction words/vector
ATmega328	32KBytes	1KBytes	2KBytes	2 instruction words/vector
ATmega328P	32KBytes	1KBytes	2KBytes	2 instruction words/vector

ATmega48A/PA/88A/PA/168A/PA/328/P support a real Read-While-Write Self-Programming mechanism. There is a separate Boot Loader Section, and the SPM instruction can only execute from there. In ATmega 48A/48PA there is no Read-While-Write support and no separate Boot Loader Section. The SPM instruction can execute from the entire Flash.

3. Resources

A comprehensive set of development tools, application notes and datasheets are available for download on <http://www.atmel.com/avr>.

4. Data Retention

Reliability Qualification results show that the projected data retention failure rate is much less than 1 PPM over 20 years at 85°C or 100 years at 25°C.

5. About Code Examples

This documentation contains simple code examples that briefly show how to use various parts of the device. These code examples assume that the part specific header file is included before compilation. Be aware that not all C compiler vendors include bit definitions in the header files and interrupt handling in C is compiler dependent. Please confirm with the C compiler documentation for more details.

For I/O Registers located in extended I/O map, “IN”, “OUT”, “SBIS”, “SBIC”, “CBI”, and “SBI” instructions must be replaced with instructions that allow access to extended I/O. Typically “LDS” and “STS” combined with “SBRS”, “SBRC”, “SBR”, and “CBR”.

6. Capacitive Touch Sensing

The Atmel® QTouch® Library provides a simple to use solution to realize touch sensitive interfaces on most Atmel AVR® microcontrollers. The QTouch Library includes support for the Atmel QTouch and Atmel QMatrix® acquisition methods.

Touch sensing can be added to any application by linking the appropriate Atmel QTouch Library for the AVR Microcontroller. This is done by using a simple set of APIs to define the touch channels and sensors, and then calling the touch sensing API's to retrieve the channel information and determine the touch sensor states.

The QTouch Library is FREE and downloadable from the Atmel website at the following location: www.atmel.com/qtouchlibrary. For implementation details and other information, refer to the [Atmel QTouch Library User Guide](#) - also available for download from Atmel website.

7. Register Summary

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Page
(0xFF)	Reserved	–	–	–	–	–	–	–	–	
(0xFE)	Reserved	–	–	–	–	–	–	–	–	
(0xFD)	Reserved	–	–	–	–	–	–	–	–	
(0xFC)	Reserved	–	–	–	–	–	–	–	–	
(0xFB)	Reserved	–	–	–	–	–	–	–	–	
(0xFA)	Reserved	–	–	–	–	–	–	–	–	
(0xF9)	Reserved	–	–	–	–	–	–	–	–	
(0xF8)	Reserved	–	–	–	–	–	–	–	–	
(0xF7)	Reserved	–	–	–	–	–	–	–	–	
(0xF6)	Reserved	–	–	–	–	–	–	–	–	
(0xF5)	Reserved	–	–	–	–	–	–	–	–	
(0xF4)	Reserved	–	–	–	–	–	–	–	–	
(0xF3)	Reserved	–	–	–	–	–	–	–	–	
(0xF2)	Reserved	–	–	–	–	–	–	–	–	
(0xF1)	Reserved	–	–	–	–	–	–	–	–	
(0xF0)	Reserved	–	–	–	–	–	–	–	–	
(0xEF)	Reserved	–	–	–	–	–	–	–	–	
(0xEE)	Reserved	–	–	–	–	–	–	–	–	
(0xED)	Reserved	–	–	–	–	–	–	–	–	
(0xEC)	Reserved	–	–	–	–	–	–	–	–	
(0xEB)	Reserved	–	–	–	–	–	–	–	–	
(0xEA)	Reserved	–	–	–	–	–	–	–	–	
(0xE9)	Reserved	–	–	–	–	–	–	–	–	
(0xE8)	Reserved	–	–	–	–	–	–	–	–	
(0xE7)	Reserved	–	–	–	–	–	–	–	–	
(0xE6)	Reserved	–	–	–	–	–	–	–	–	
(0xE5)	Reserved	–	–	–	–	–	–	–	–	
(0xE4)	Reserved	–	–	–	–	–	–	–	–	
(0xE3)	Reserved	–	–	–	–	–	–	–	–	
(0xE2)	Reserved	–	–	–	–	–	–	–	–	
(0xE1)	Reserved	–	–	–	–	–	–	–	–	
(0xE0)	Reserved	–	–	–	–	–	–	–	–	
(0xDF)	Reserved	–	–	–	–	–	–	–	–	
(0xDE)	Reserved	–	–	–	–	–	–	–	–	
(0xDD)	Reserved	–	–	–	–	–	–	–	–	
(0xDC)	Reserved	–	–	–	–	–	–	–	–	
(0xDB)	Reserved	–	–	–	–	–	–	–	–	
(0xDA)	Reserved	–	–	–	–	–	–	–	–	
(0xD9)	Reserved	–	–	–	–	–	–	–	–	
(0xD8)	Reserved	–	–	–	–	–	–	–	–	
(0xD7)	Reserved	–	–	–	–	–	–	–	–	
(0xD6)	Reserved	–	–	–	–	–	–	–	–	
(0xD5)	Reserved	–	–	–	–	–	–	–	–	
(0xD4)	Reserved	–	–	–	–	–	–	–	–	
(0xD3)	Reserved	–	–	–	–	–	–	–	–	
(0xD2)	Reserved	–	–	–	–	–	–	–	–	
(0xD1)	Reserved	–	–	–	–	–	–	–	–	
(0xD0)	Reserved	–	–	–	–	–	–	–	–	
(0xCF)	Reserved	–	–	–	–	–	–	–	–	
(0xCE)	Reserved	–	–	–	–	–	–	–	–	
(0xCD)	Reserved	–	–	–	–	–	–	–	–	
(0xCC)	Reserved	–	–	–	–	–	–	–	–	
(0xCB)	Reserved	–	–	–	–	–	–	–	–	
(0xCA)	Reserved	–	–	–	–	–	–	–	–	
(0xC9)	Reserved	–	–	–	–	–	–	–	–	
(0xC8)	Reserved	–	–	–	–	–	–	–	–	
(0xC7)	Reserved	–	–	–	–	–	–	–	–	
(0xC6)	UDR0	USART I/O Data Register								201
(0xC5)	UBRR0H	USART Baud Rate Register High								205
(0xC4)	UBRR0L	USART Baud Rate Register Low								205
(0xC3)	Reserved	–	–	–	–	–	–	–	–	
(0xC2)	UCSR0C	UMSEL01	UMSEL00	UPM01	UPM00	USBS0	UCSZ01 / UDORD0	UCSZ00 / UCPHA0	UCPOL0	203/214
(0xC1)	UCSR0B	RXCIE0	TXCIE0	UDRIE0	RXEN0	TXEN0	UCSZ02	RXB80	TXB80	202
(0xC0)	UCSR0A	RXC0	TXC0	UDRE0	FE0	DOR0	UPE0	U2X0	MPCM0	201



ATmega48A/PA/88A/PA/168A/PA/328/P

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Page	
(0xBF)	Reserved	–	–	–	–	–	–	–	–		
(0xBE)	Reserved	–	–	–	–	–	–	–	–		
(0xBD)	TWAMR	TWAM6	TWAM5	TWAM4	TWAM3	TWAM2	TWAM1	TWAM0	–	246	
(0xBC)	TWCR	TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	–	TWIE	243	
(0xBB)	TWDR	2-wire Serial Interface Data Register									245
(0xBA)	TWAR	TWA6	TWA5	TWA4	TWA3	TWA2	TWA1	TWA0	TWGCE	246	
(0xB9)	TWSR	TWS7	TWS6	TWS5	TWS4	TWS3	–	TWPS1	TWPS0	245	
(0xB8)	TWBR	2-wire Serial Interface Bit Rate Register									243
(0xB7)	Reserved	–	–	–	–	–	–	–	–		
(0xB6)	ASSR	–	EXCLK	AS2	TCN2UB	OCR2AUB	OCR2BUB	TCR2AUB	TCR2BUB	166	
(0xB5)	Reserved	–	–	–	–	–	–	–	–		
(0xB4)	OCR2B	Timer/Counter2 Output Compare Register B									164
(0xB3)	OCR2A	Timer/Counter2 Output Compare Register A									164
(0xB2)	TCNT2	Timer/Counter2 (8-bit)									164
(0xB1)	TCCR2B	FOC2A	FOC2B	–	–	WGM22	CS22	CS21	CS20	163	
(0xB0)	TCCR2A	COM2A1	COM2A0	COM2B1	COM2B0	–	–	WGM21	WGM20	160	
(0xAF)	Reserved	–	–	–	–	–	–	–	–		
(0xAE)	Reserved	–	–	–	–	–	–	–	–		
(0xAD)	Reserved	–	–	–	–	–	–	–	–		
(0xAC)	Reserved	–	–	–	–	–	–	–	–		
(0xAB)	Reserved	–	–	–	–	–	–	–	–		
(0xAA)	Reserved	–	–	–	–	–	–	–	–		
(0xA9)	Reserved	–	–	–	–	–	–	–	–		
(0xA8)	Reserved	–	–	–	–	–	–	–	–		
(0xA7)	Reserved	–	–	–	–	–	–	–	–		
(0xA6)	Reserved	–	–	–	–	–	–	–	–		
(0xA5)	Reserved	–	–	–	–	–	–	–	–		
(0xA4)	Reserved	–	–	–	–	–	–	–	–		
(0xA3)	Reserved	–	–	–	–	–	–	–	–		
(0xA2)	Reserved	–	–	–	–	–	–	–	–		
(0xA1)	Reserved	–	–	–	–	–	–	–	–		
(0xA0)	Reserved	–	–	–	–	–	–	–	–		
(0x9F)	Reserved	–	–	–	–	–	–	–	–		
(0x9E)	Reserved	–	–	–	–	–	–	–	–		
(0x9D)	Reserved	–	–	–	–	–	–	–	–		
(0x9C)	Reserved	–	–	–	–	–	–	–	–		
(0x9B)	Reserved	–	–	–	–	–	–	–	–		
(0x9A)	Reserved	–	–	–	–	–	–	–	–		
(0x99)	Reserved	–	–	–	–	–	–	–	–		
(0x98)	Reserved	–	–	–	–	–	–	–	–		
(0x97)	Reserved	–	–	–	–	–	–	–	–		
(0x96)	Reserved	–	–	–	–	–	–	–	–		
(0x95)	Reserved	–	–	–	–	–	–	–	–		
(0x94)	Reserved	–	–	–	–	–	–	–	–		
(0x93)	Reserved	–	–	–	–	–	–	–	–		
(0x92)	Reserved	–	–	–	–	–	–	–	–		
(0x91)	Reserved	–	–	–	–	–	–	–	–		
(0x90)	Reserved	–	–	–	–	–	–	–	–		
(0x8F)	Reserved	–	–	–	–	–	–	–	–		
(0x8E)	Reserved	–	–	–	–	–	–	–	–		
(0x8D)	Reserved	–	–	–	–	–	–	–	–		
(0x8C)	Reserved	–	–	–	–	–	–	–	–		
(0x8B)	OCR1BH	Timer/Counter1 - Output Compare Register B High Byte									140
(0x8A)	OCR1BL	Timer/Counter1 - Output Compare Register B Low Byte									140
(0x89)	OCR1AH	Timer/Counter1 - Output Compare Register A High Byte									140
(0x88)	OCR1AL	Timer/Counter1 - Output Compare Register A Low Byte									140
(0x87)	ICR1H	Timer/Counter1 - Input Capture Register High Byte									140
(0x86)	ICR1L	Timer/Counter1 - Input Capture Register Low Byte									140
(0x85)	TCNT1H	Timer/Counter1 - Counter Register High Byte									140
(0x84)	TCNT1L	Timer/Counter1 - Counter Register Low Byte									140
(0x83)	Reserved	–	–	–	–	–	–	–	–		
(0x82)	TCCR1C	FOC1A	FOC1B	–	–	–	–	–	–	139	
(0x81)	TCCR1B	ICNC1	ICES1	–	WGM13	WGM12	CS12	CS11	CS10	138	
(0x80)	TCCR1A	COM1A1	COM1A0	COM1B1	COM1B0	–	–	WGM11	WGM10	136	
(0x7F)	DIDR1	–	–	–	–	–	–	AIN1D	AIN0D	251	
(0x7E)	DIDR0	–	–	ADC5D	ADC4D	ADC3D	ADC2D	ADC1D	ADC0D	268	



ATmega48A/PA/88A/PA/168A/PA/328/P

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Page
(0x7D)	Reserved	–	–	–	–	–	–	–	–	
(0x7C)	ADMUX	REFS1	REFS0	ADLAR	–	MUX3	MUX2	MUX1	MUX0	264
(0x7B)	ADCSRB	–	ACME	–	–	–	ADTS2	ADTS1	ADTS0	267
(0x7A)	ADCSRA	ADEN	ADSC	ADATE	ADIF	ADIE	ADPS2	ADPS1	ADPS0	265
(0x79)	ADCH	ADC Data Register High byte								267
(0x78)	ADCL	ADC Data Register Low byte								267
(0x77)	Reserved	–	–	–	–	–	–	–	–	
(0x76)	Reserved	–	–	–	–	–	–	–	–	
(0x75)	Reserved	–	–	–	–	–	–	–	–	
(0x74)	Reserved	–	–	–	–	–	–	–	–	
(0x73)	Reserved	–	–	–	–	–	–	–	–	
(0x72)	Reserved	–	–	–	–	–	–	–	–	
(0x71)	Reserved	–	–	–	–	–	–	–	–	
(0x70)	TIMSK2	–	–	–	–	–	OCIE2B	OCIE2A	TOIE2	165
(0x6F)	TIMSK1	–	–	ICIE1	–	–	OCIE1B	OCIE1A	TOIE1	141
(0x6E)	TIMSK0	–	–	–	–	–	OCIE0B	OCIE0A	TOIE0	113
(0x6D)	PCMSK2	PCINT23	PCINT22	PCINT21	PCINT20	PCINT19	PCINT18	PCINT17	PCINT16	76
(0x6C)	PCMSK1	–	PCINT14	PCINT13	PCINT12	PCINT11	PCINT10	PCINT9	PCINT8	76
(0x6B)	PCMSK0	PCINT7	PCINT6	PCINT5	PCINT4	PCINT3	PCINT2	PCINT1	PCINT0	76
(0x6A)	Reserved	–	–	–	–	–	–	–	–	
(0x69)	EICRA	–	–	–	–	ISC11	ISC10	ISC01	ISC00	73
(0x68)	PCICR	–	–	–	–	–	PCIE2	PCIE1	PCIE0	
(0x67)	Reserved	–	–	–	–	–	–	–	–	
(0x66)	OSCCAL	Oscillator Calibration Register								38
(0x65)	Reserved	–	–	–	–	–	–	–	–	
(0x64)	PRR	PRTWI	PRTIM2	PRTIM0	–	PRTIM1	PRSPI	PRUSART0	PRADC	43
(0x63)	Reserved	–	–	–	–	–	–	–	–	
(0x62)	Reserved	–	–	–	–	–	–	–	–	
(0x61)	CLKPR	CLKPCE	–	–	–	CLKPS3	CLKPS2	CLKPS1	CLKPS0	38
(0x60)	WDTCR	WDIF	WDIE	WDP3	WDCE	WDE	WDP2	WDP1	WDP0	56
0x3F (0x5F)	SREG	I	T	H	S	V	N	Z	C	10
0x3E (0x5E)	SPH	–	–	–	–	–	(SP10) ⁵	SP9	SP8	13
0x3D (0x5D)	SPL	SP7	SP6	SP5	SP4	SP3	SP2	SP1	SP0	13
0x3C (0x5C)	Reserved	–	–	–	–	–	–	–	–	
0x3B (0x5B)	Reserved	–	–	–	–	–	–	–	–	
0x3A (0x5A)	Reserved	–	–	–	–	–	–	–	–	
0x39 (0x59)	Reserved	–	–	–	–	–	–	–	–	
0x38 (0x58)	Reserved	–	–	–	–	–	–	–	–	
0x37 (0x57)	SPMCSR	SPMIE	(RWWSB) ⁵	–	(RWWRE) ⁵	BLBSET	PGWRT	PGERS	SELFPRGEN	295
0x36 (0x56)	Reserved	–	–	–	–	–	–	–	–	
0x35 (0x55)	MCUCR	–	BODS ⁽⁶⁾	BODSE ⁽⁶⁾	PUD	–	–	IVSEL	IVCE	46/70/94
0x34 (0x54)	MCUSR	–	–	–	–	WDRF	BORF	EXTRF	PORF	56
0x33 (0x53)	SMCR	–	–	–	–	SM2	SM1	SM0	SE	41
0x32 (0x52)	Reserved	–	–	–	–	–	–	–	–	
0x31 (0x51)	Reserved	–	–	–	–	–	–	–	–	
0x30 (0x50)	ACSR	ACD	ACBG	ACO	ACI	ACIE	ACIC	ACIS1	ACIS0	249
0x2F (0x4F)	Reserved	–	–	–	–	–	–	–	–	
0x2E (0x4E)	SPDR	SPI Data Register								177
0x2D (0x4D)	SPSR	SPIF	WCOL	–	–	–	–	–	SPI2X	176
0x2C (0x4C)	SPCR	SPIE	SPE	DORD	MSTR	CPOL	CPHA	SPR1	SPR0	175
0x2B (0x4B)	GPOR2	General Purpose I/O Register 2								26
0x2A (0x4A)	GPOR1	General Purpose I/O Register 1								26
0x29 (0x49)	Reserved	–	–	–	–	–	–	–	–	
0x28 (0x48)	OCR0B	Timer/Counter0 Output Compare Register B								
0x27 (0x47)	OCR0A	Timer/Counter0 Output Compare Register A								
0x26 (0x46)	TCNT0	Timer/Counter0 (8-bit)								
0x25 (0x45)	TCCR0B	FOC0A	FOC0B	–	–	WGM02	CS02	CS01	CS00	
0x24 (0x44)	TCCR0A	COM0A1	COM0A0	COM0B1	COM0B0	–	–	WGM01	WGM00	
0x23 (0x43)	GTCCR	TSM	–	–	–	–	–	PSRASY	PSRSYNC	145/167
0x22 (0x42)	EEARH	(EEPROM Address Register High Byte) ⁵								22
0x21 (0x41)	EEARL	EEPROM Address Register Low Byte								22
0x20 (0x40)	EEDR	EEPROM Data Register								22
0x1F (0x3F)	EEDR	–	–	EEDR1	EEDR0	EERIE	EEMPE	EEPE	EERE	22
0x1E (0x3E)	GPOR0	General Purpose I/O Register 0								26
0x1D (0x3D)	EIMSK	–	–	–	–	–	–	INT1	INT0	74
0x1C (0x3C)	EIFR	–	–	–	–	–	–	INTF1	INTF0	74



Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Page
0x1B (0x3B)	PCIFR	–	–	–	–	–	PCIF2	PCIF1	PCIF0	
0x1A (0x3A)	Reserved	–	–	–	–	–	–	–	–	
0x19 (0x39)	Reserved	–	–	–	–	–	–	–	–	
0x18 (0x38)	Reserved	–	–	–	–	–	–	–	–	
0x17 (0x37)	TIFR2	–	–	–	–	–	OCF2B	OCF2A	TOV2	165
0x16 (0x36)	TIFR1	–	–	ICF1	–	–	OCF1B	OCF1A	TOV1	141
0x15 (0x35)	TIFR0	–	–	–	–	–	OCF0B	OCF0A	TOV0	
0x14 (0x34)	Reserved	–	–	–	–	–	–	–	–	
0x13 (0x33)	Reserved	–	–	–	–	–	–	–	–	
0x12 (0x32)	Reserved	–	–	–	–	–	–	–	–	
0x11 (0x31)	Reserved	–	–	–	–	–	–	–	–	
0x10 (0x30)	Reserved	–	–	–	–	–	–	–	–	
0x0F (0x2F)	Reserved	–	–	–	–	–	–	–	–	
0x0E (0x2E)	Reserved	–	–	–	–	–	–	–	–	
0x0D (0x2D)	Reserved	–	–	–	–	–	–	–	–	
0x0C (0x2C)	Reserved	–	–	–	–	–	–	–	–	
0x0B (0x2B)	PORTD	PORTD7	PORTD6	PORTD5	PORTD4	PORTD3	PORTD2	PORTD1	PORTD0	95
0x0A (0x2A)	DDRD	DDD7	DDD6	DDD5	DDD4	DDD3	DDD2	DDD1	DDD0	95
0x09 (0x29)	PIND	PIND7	PIND6	PIND5	PIND4	PIND3	PIND2	PIND1	PIND0	95
0x08 (0x28)	PORTC	–	PORTC6	PORTC5	PORTC4	PORTC3	PORTC2	PORTC1	PORTC0	94
0x07 (0x27)	DDRC	–	DDC6	DDC5	DDC4	DDC3	DDC2	DDC1	DDC0	94
0x06 (0x26)	PINC	–	PINC6	PINC5	PINC4	PINC3	PINC2	PINC1	PINC0	94
0x05 (0x25)	PORTB	PORTB7	PORTB6	PORTB5	PORTB4	PORTB3	PORTB2	PORTB1	PORTB0	94
0x04 (0x24)	DDRB	DDB7	DDB6	DDB5	DDB4	DDB3	DDB2	DDB1	DDB0	94
0x03 (0x23)	PINB	PINB7	PINB6	PINB5	PINB4	PINB3	PINB2	PINB1	PINB0	94
0x02 (0x22)	Reserved	–	–	–	–	–	–	–	–	
0x01 (0x21)	Reserved	–	–	–	–	–	–	–	–	
0x0 (0x20)	Reserved	–	–	–	–	–	–	–	–	

- Note:
- For compatibility with future devices, reserved bits should be written to zero if accessed. Reserved I/O memory addresses should never be written.
 - I/O Registers within the address range 0x00 - 0x1F are directly bit-accessible using the SBI and CBI instructions. In these registers, the value of single bits can be checked by using the SBIS and SBIC instructions.
 - Some of the Status Flags are cleared by writing a logical one to them. Note that, unlike most other AVRs, the CBI and SBI instructions will only operate on the specified bit, and can therefore be used on registers containing such Status Flags. The CBI and SBI instructions work with registers 0x00 to 0x1F only.
 - When using the I/O specific commands IN and OUT, the I/O addresses 0x00 - 0x3F must be used. When addressing I/O Registers as data space using LD and ST instructions, 0x20 must be added to these addresses. The ATmega48A/PA/88A/PA/168A/PA/328/P is a complex microcontroller with more peripheral units than can be supported within the 64 location reserved in Opcode for the IN and OUT instructions. For the Extended I/O space from 0x60 - 0xFF in SRAM, only the ST/STS/STD and LD/LDS/LDD instructions can be used.
 - Only valid for ATmega88A/88PA/168A/168PA/328/328P.
 - BODS and BODSE only available for picoPower devices ATmega48PA/88PA/168PA/328P



8. Instruction Set Summary

Mnemonics	Operands	Description	Operation	Flags	#Clocks
ARITHMETIC AND LOGIC INSTRUCTIONS					
ADD	Rd, Rr	Add two Registers	$Rd \leftarrow Rd + Rr$	Z,C,N,V,H	1
ADC	Rd, Rr	Add with Carry two Registers	$Rd \leftarrow Rd + Rr + C$	Z,C,N,V,H	1
ADIW	RdI,K	Add Immediate to Word	$Rdh:Rdl \leftarrow Rdh:Rdl + K$	Z,C,N,V,S	2
SUB	Rd, Rr	Subtract two Registers	$Rd \leftarrow Rd - Rr$	Z,C,N,V,H	1
SUBI	Rd, K	Subtract Constant from Register	$Rd \leftarrow Rd - K$	Z,C,N,V,H	1
SBC	Rd, Rr	Subtract with Carry two Registers	$Rd \leftarrow Rd - Rr - C$	Z,C,N,V,H	1
SBCI	Rd, K	Subtract with Carry Constant from Reg.	$Rd \leftarrow Rd - K - C$	Z,C,N,V,H	1
SBIW	RdI,K	Subtract Immediate from Word	$Rdh:Rdl \leftarrow Rdh:Rdl - K$	Z,C,N,V,S	2
AND	Rd, Rr	Logical AND Registers	$Rd \leftarrow Rd \bullet Rr$	Z,N,V	1
ANDI	Rd, K	Logical AND Register and Constant	$Rd \leftarrow Rd \bullet K$	Z,N,V	1
OR	Rd, Rr	Logical OR Registers	$Rd \leftarrow Rd \vee Rr$	Z,N,V	1
ORI	Rd, K	Logical OR Register and Constant	$Rd \leftarrow Rd \vee K$	Z,N,V	1
EOR	Rd, Rr	Exclusive OR Registers	$Rd \leftarrow Rd \oplus Rr$	Z,N,V	1
COM	Rd	One's Complement	$Rd \leftarrow 0xFF - Rd$	Z,C,N,V	1
NEG	Rd	Two's Complement	$Rd \leftarrow 0x00 - Rd$	Z,C,N,V,H	1
SBR	Rd,K	Set Bit(s) in Register	$Rd \leftarrow Rd \vee K$	Z,N,V	1
CBR	Rd,K	Clear Bit(s) in Register	$Rd \leftarrow Rd \bullet (0xFF - K)$	Z,N,V	1
INC	Rd	Increment	$Rd \leftarrow Rd + 1$	Z,N,V	1
DEC	Rd	Decrement	$Rd \leftarrow Rd - 1$	Z,N,V	1
TST	Rd	Test for Zero or Minus	$Rd \leftarrow Rd \bullet Rd$	Z,N,V	1
CLR	Rd	Clear Register	$Rd \leftarrow Rd \oplus Rd$	Z,N,V	1
SER	Rd	Set Register	$Rd \leftarrow 0xFF$	None	1
MUL	Rd, Rr	Multiply Unsigned	$R1:R0 \leftarrow Rd \times Rr$	Z,C	2
MULS	Rd, Rr	Multiply Signed	$R1:R0 \leftarrow Rd \times Rr$	Z,C	2
MULSU	Rd, Rr	Multiply Signed with Unsigned	$R1:R0 \leftarrow Rd \times Rr$	Z,C	2
FMUL	Rd, Rr	Fractional Multiply Unsigned	$R1:R0 \leftarrow (Rd \times Rr) \lll 1$	Z,C	2
FMULS	Rd, Rr	Fractional Multiply Signed	$R1:R0 \leftarrow (Rd \times Rr) \lll 1$	Z,C	2
FMULSU	Rd, Rr	Fractional Multiply Signed with Unsigned	$R1:R0 \leftarrow (Rd \times Rr) \lll 1$	Z,C	2
BRANCH INSTRUCTIONS					
RJMP	k	Relative Jump	$PC \leftarrow PC + k + 1$	None	2
IJMP		Indirect Jump to (Z)	$PC \leftarrow Z$	None	2
JMP ⁽¹⁾	k	Direct Jump	$PC \leftarrow k$	None	3
RCALL	k	Relative Subroutine Call	$PC \leftarrow PC + k + 1$	None	3
ICALL		Indirect Call to (Z)	$PC \leftarrow Z$	None	3
CALL ⁽¹⁾	k	Direct Subroutine Call	$PC \leftarrow k$	None	4
RET		Subroutine Return	$PC \leftarrow STACK$	None	4
RETI		Interrupt Return	$PC \leftarrow STACK$	I	4
CPSE	Rd,Rr	Compare, Skip if Equal	if $(Rd = Rr)$ $PC \leftarrow PC + 2$ or 3	None	1/2/3
CP	Rd,Rr	Compare	$Rd - Rr$	Z, N, V, C, H	1
CPC	Rd,Rr	Compare with Carry	$Rd - Rr - C$	Z, N, V, C, H	1
CPI	Rd,K	Compare Register with Immediate	$Rd - K$	Z, N, V, C, H	1
SBRC	Rr, b	Skip if Bit in Register Cleared	if $(Rr(b)=0)$ $PC \leftarrow PC + 2$ or 3	None	1/2/3
SBRS	Rr, b	Skip if Bit in Register is Set	if $(Rr(b)=1)$ $PC \leftarrow PC + 2$ or 3	None	1/2/3
SBIC	P, b	Skip if Bit in I/O Register Cleared	if $(P(b)=0)$ $PC \leftarrow PC + 2$ or 3	None	1/2/3
SBIS	P, b	Skip if Bit in I/O Register is Set	if $(P(b)=1)$ $PC \leftarrow PC + 2$ or 3	None	1/2/3
BRBS	s, k	Branch if Status Flag Set	if $(SREG(s) = 1)$ then $PC \leftarrow PC + k + 1$	None	1/2
BRBC	s, k	Branch if Status Flag Cleared	if $(SREG(s) = 0)$ then $PC \leftarrow PC + k + 1$	None	1/2
BREQ	k	Branch if Equal	if $(Z = 1)$ then $PC \leftarrow PC + k + 1$	None	1/2
BRNE	k	Branch if Not Equal	if $(Z = 0)$ then $PC \leftarrow PC + k + 1$	None	1/2
BRCS	k	Branch if Carry Set	if $(C = 1)$ then $PC \leftarrow PC + k + 1$	None	1/2
BRCC	k	Branch if Carry Cleared	if $(C = 0)$ then $PC \leftarrow PC + k + 1$	None	1/2
BRSH	k	Branch if Same or Higher	if $(C = 0)$ then $PC \leftarrow PC + k + 1$	None	1/2
BRLO	k	Branch if Lower	if $(C = 1)$ then $PC \leftarrow PC + k + 1$	None	1/2
BRMI	k	Branch if Minus	if $(N = 1)$ then $PC \leftarrow PC + k + 1$	None	1/2
BRPL	k	Branch if Plus	if $(N = 0)$ then $PC \leftarrow PC + k + 1$	None	1/2
BRGE	k	Branch if Greater or Equal, Signed	if $(N \oplus V = 0)$ then $PC \leftarrow PC + k + 1$	None	1/2
BRLT	k	Branch if Less Than Zero, Signed	if $(N \oplus V = 1)$ then $PC \leftarrow PC + k + 1$	None	1/2
BRHS	k	Branch if Half Carry Flag Set	if $(H = 1)$ then $PC \leftarrow PC + k + 1$	None	1/2
BRHC	k	Branch if Half Carry Flag Cleared	if $(H = 0)$ then $PC \leftarrow PC + k + 1$	None	1/2
BRTS	k	Branch if T Flag Set	if $(T = 1)$ then $PC \leftarrow PC + k + 1$	None	1/2
BRTC	k	Branch if T Flag Cleared	if $(T = 0)$ then $PC \leftarrow PC + k + 1$	None	1/2
BRVS	k	Branch if Overflow Flag is Set	if $(V = 1)$ then $PC \leftarrow PC + k + 1$	None	1/2
BRVC	k	Branch if Overflow Flag is Cleared	if $(V = 0)$ then $PC \leftarrow PC + k + 1$	None	1/2

ATmega48A/PA/88A/PA/168A/PA/328/P

Mnemonics	Operands	Description	Operation	Flags	#Clocks
BRIE	k	Branch if Interrupt Enabled	if (I = 1) then PC ← PC + k + 1	None	1/2
BRID	k	Branch if Interrupt Disabled	if (I = 0) then PC ← PC + k + 1	None	1/2
BIT AND BIT-TEST INSTRUCTIONS					
SBI	P,b	Set Bit in I/O Register	I/O(P,b) ← 1	None	2
CBI	P,b	Clear Bit in I/O Register	I/O(P,b) ← 0	None	2
LSL	Rd	Logical Shift Left	Rd(n+1) ← Rd(n), Rd(0) ← 0	Z,C,N,V	1
LSR	Rd	Logical Shift Right	Rd(n) ← Rd(n+1), Rd(7) ← 0	Z,C,N,V	1
ROL	Rd	Rotate Left Through Carry	Rd(0) ← C, Rd(n+1) ← Rd(n), C ← Rd(7)	Z,C,N,V	1
ROR	Rd	Rotate Right Through Carry	Rd(7) ← C, Rd(n) ← Rd(n+1), C ← Rd(0)	Z,C,N,V	1
ASR	Rd	Arithmetic Shift Right	Rd(n) ← Rd(n+1), n=0..6	Z,C,N,V	1
SWAP	Rd	Swap Nibbles	Rd(3..0) ← Rd(7..4), Rd(7..4) ← Rd(3..0)	None	1
BSET	s	Flag Set	SREG(s) ← 1	SREG(s)	1
BCLR	s	Flag Clear	SREG(s) ← 0	SREG(s)	1
BST	Rr, b	Bit Store from Register to T	T ← Rr(b)	T	1
BLD	Rd, b	Bit load from T to Register	Rd(b) ← T	None	1
SEC		Set Carry	C ← 1	C	1
CLC		Clear Carry	C ← 0	C	1
SEN		Set Negative Flag	N ← 1	N	1
CLN		Clear Negative Flag	N ← 0	N	1
SEZ		Set Zero Flag	Z ← 1	Z	1
CLZ		Clear Zero Flag	Z ← 0	Z	1
SEI		Global Interrupt Enable	I ← 1	I	1
CLI		Global Interrupt Disable	I ← 0	I	1
SES		Set Signed Test Flag	S ← 1	S	1
CLS		Clear Signed Test Flag	S ← 0	S	1
SEV		Set Twos Complement Overflow.	V ← 1	V	1
CLV		Clear Twos Complement Overflow	V ← 0	V	1
SET		Set T in SREG	T ← 1	T	1
CLT		Clear T in SREG	T ← 0	T	1
SEH		Set Half Carry Flag in SREG	H ← 1	H	1
CLH		Clear Half Carry Flag in SREG	H ← 0	H	1
DATA TRANSFER INSTRUCTIONS					
MOV	Rd, Rr	Move Between Registers	Rd ← Rr	None	1
MOVW	Rd, Rr	Copy Register Word	Rd+1:Rd ← Rr+1:Rr	None	1
LDI	Rd, K	Load Immediate	Rd ← K	None	1
LD	Rd, X	Load Indirect	Rd ← (X)	None	2
LD	Rd, X+	Load Indirect and Post-Inc.	Rd ← (X), X ← X + 1	None	2
LD	Rd, -X	Load Indirect and Pre-Dec.	X ← X - 1, Rd ← (X)	None	2
LD	Rd, Y	Load Indirect	Rd ← (Y)	None	2
LD	Rd, Y+	Load Indirect and Post-Inc.	Rd ← (Y), Y ← Y + 1	None	2
LD	Rd, -Y	Load Indirect and Pre-Dec.	Y ← Y - 1, Rd ← (Y)	None	2
LDD	Rd, Y+q	Load Indirect with Displacement	Rd ← (Y + q)	None	2
LD	Rd, Z	Load Indirect	Rd ← (Z)	None	2
LD	Rd, Z+	Load Indirect and Post-Inc.	Rd ← (Z), Z ← Z+1	None	2
LD	Rd, -Z	Load Indirect and Pre-Dec.	Z ← Z - 1, Rd ← (Z)	None	2
LDD	Rd, Z+q	Load Indirect with Displacement	Rd ← (Z + q)	None	2
LDS	Rd, k	Load Direct from SRAM	Rd ← (k)	None	2
ST	X, Rr	Store Indirect	(X) ← Rr	None	2
ST	X+, Rr	Store Indirect and Post-Inc.	(X) ← Rr, X ← X + 1	None	2
ST	-X, Rr	Store Indirect and Pre-Dec.	X ← X - 1, (X) ← Rr	None	2
ST	Y, Rr	Store Indirect	(Y) ← Rr	None	2
ST	Y+, Rr	Store Indirect and Post-Inc.	(Y) ← Rr, Y ← Y + 1	None	2
ST	-Y, Rr	Store Indirect and Pre-Dec.	Y ← Y - 1, (Y) ← Rr	None	2
STD	Y+q, Rr	Store Indirect with Displacement	(Y + q) ← Rr	None	2
ST	Z, Rr	Store Indirect	(Z) ← Rr	None	2
ST	Z+, Rr	Store Indirect and Post-Inc.	(Z) ← Rr, Z ← Z + 1	None	2
ST	-Z, Rr	Store Indirect and Pre-Dec.	Z ← Z - 1, (Z) ← Rr	None	2
STD	Z+q, Rr	Store Indirect with Displacement	(Z + q) ← Rr	None	2
STS	k, Rr	Store Direct to SRAM	(k) ← Rr	None	2
LPM		Load Program Memory	R0 ← (Z)	None	3
LPM	Rd, Z	Load Program Memory	Rd ← (Z)	None	3
LPM	Rd, Z+	Load Program Memory and Post-Inc	Rd ← (Z), Z ← Z+1	None	3
SPM		Store Program Memory	(Z) ← R1:R0	None	-
IN	Rd, P	In Port	Rd ← P	None	1
OUT	P, Rr	Out Port	P ← Rr	None	1
PUSH	Rr	Push Register on Stack	STACK ← Rr	None	2



Mnemonics	Operands	Description	Operation	Flags	#Clocks
POP	Rd	Pop Register from Stack	Rd ← STACK	None	2
MCU CONTROL INSTRUCTIONS					
NOP		No Operation		None	1
SLEEP		Sleep	(see specific descr. for Sleep function)	None	1
WDR		Watchdog Reset	(see specific descr. for WDR/timer)	None	1
BREAK		Break	For On-chip Debug Only	None	N/A

Note: 1. These instructions are only available in ATmega168PA and ATmega328P.

9. Ordering Information

9.1 ATmega48A

Speed (MHz)	Power Supply (V)	Ordering Code ⁽²⁾	Package ⁽¹⁾	Operational Range
20 ⁽³⁾	1.8 - 5.5	ATmega48A-AU ATmega48A-AUR ⁽⁵⁾ ATmega48A-CCU ATmega48A-CCUR ⁽⁵⁾ ATmega48A-MMH ⁽⁴⁾ ATmega48A-MMHR ⁽⁴⁾⁽⁵⁾ ATmega48A-MU ATmega48A-MUR ⁽⁵⁾ ATmega48A-PU	32A 32A 32CC1 32CC1 28M1 28M1 32M1-A 32M1-A 28P3	Industrial (-40°C to 85°C)

- Note:
1. This device can also be supplied in wafer form. Please contact your local Atmel sales office for detailed ordering information and minimum quantities.
 2. Pb-free packaging complies to the European Directive for Restriction of Hazardous Substances (RoHS directive). Also Halide free and fully Green.
 3. See "[Speed Grades](#)" on page 322.
 4. NiPdAu Lead Finish.
 5. Tape & Reel.

Package Type	
32A	32-lead, Thin (1.0 mm) Plastic Quad Flat Package (TQFP)
32CC1	32-ball, 4 x 4 x 0.6 mm package, ball pitch 0.5 mm, Ultra Thin, Fine-Pitch Ball Grill Array (UFBGA)
28M1	28-pad, 4 x 4 x 1.0 body, Lead Pitch 0.45 mm Quad Flat No-Lead/Micro Lead Frame Package (QFN/MLF)
32M1-A	32-pad, 5 x 5 x 1.0 body, Lead Pitch 0.50 mm Quad Flat No-Lead/Micro Lead Frame Package (QFN/MLF)
28P3	28-lead, 0.300" Wide, Plastic Dual Inline Package (PDIP)

9.2 ATmega48PA

Speed (MHz) ⁽³⁾	Power Supply	Ordering Code ⁽²⁾	Package ⁽¹⁾	Operational Range
20	1.8 - 5.5	ATmega48PA-AU	32A	Industrial (-40°C to 85°C)
		ATmega48PA-AUR ⁽⁵⁾	32A	
		ATmega48PA-CCU	32CC1	
		ATmega48PA-CCUR ⁽⁵⁾	32CC1	
		ATmega48PA-MMH ⁽⁴⁾	28M1	
		ATmega48PA-MMHR ⁽⁴⁾⁽⁵⁾	28M1	
		ATmega48PA-MU	32M1-A	
		ATmega48PA-MUR ⁽⁵⁾	32M1-A	
		ATmega48PA-PU	28P3	Industrial (-40°C to 105°C)
		ATmega48PA-AN	32A	
		ATmega48PA-ANR ⁽⁴⁾	32A	
		ATmega48PA-MMN	28M1	
		ATmega48PA-MMNR ⁽⁴⁾	28M1	
		ATmega48PA-MN	32M1-A	
ATmega48PA-MNR ⁽⁴⁾	32M1-A			
ATmega48PA-PN	28P3			

- Note:
1. This device can also be supplied in wafer form. Please contact your local Atmel sales office for detailed ordering information and minimum quantities.
 2. Pb-free packaging complies to the European Directive for Restriction of Hazardous Substances (RoHS directive). Also Halide free and fully Green.
 3. See "[Speed Grades](#)" on page 322.
 4. NiPdAu Lead Finish.
 5. Tape & Reel.

Package Type	
32A	32-lead, Thin (1.0mm) Plastic Quad Flat Package (TQFP)
32CC1	32-ball, 4 x 4 x 0.6mm package, ball pitch 0.5mm, Ultra Thin, Fine-Pitch Ball Grill Array (UFBGA)
28M1	28-pad, 4 x 4 x 1.0 body, Lead Pitch 0.45mm Quad Flat No-Lead/Micro Lead Frame Package (QFN/MLF)
32M1-A	32-pad, 5 x 5 x 1.0 body, Lead Pitch 0.50mm Quad Flat No-Lead/Micro Lead Frame Package (QFN/MLF)
28P3	28-lead, 0.300" Wide, Plastic Dual Inline Package (PDIP)



9.3 ATmega88A

Speed (MHz)	Power Supply (V)	Ordering Code ⁽²⁾	Package ⁽¹⁾	Operational Range
20 ⁽³⁾	1.8 - 5.5	ATmega88A-AU ATmega88A-AUR ⁽⁵⁾ ATmega88A-CCU ATmega88A-CCUR ⁽⁵⁾ ATmega88A-MMH ⁽⁴⁾ ATmega88A-MMHR ⁽⁴⁾⁽⁵⁾ ATmega88A-MU ATmega88A-MUR ⁽⁵⁾ ATmega88A-PU	32A 32A 32CC1 32CC1 28M1 28M1 32M1-A 32M1-A 28P3	Industrial (-40°C to 85°C)

- Note:
1. This device can also be supplied in wafer form. Please contact your local Atmel sales office for detailed ordering information and minimum quantities.
 2. Pb-free packaging complies to the European Directive for Restriction of Hazardous Substances (RoHS directive). Also Halide free and fully Green.
 3. See "[Speed Grades](#)" on page 322.
 4. NiPdAu Lead Finish.
 5. Tape & Reel.

Package Type	
32A	32-lead, Thin (1.0mm) Plastic Quad Flat Package (TQFP)
32CC1	32-ball, 4 x 4 x 0.6mm package, ball pitch 0.5mm, Ultra Thin, Fine-Pitch Ball Grill Array (UFBGA)
28M1	28-pad, 4 x 4 x 1.0 body, Lead Pitch 0.45mm Quad Flat No-Lead/Micro Lead Frame Package (QFN/MLF)
32M1-A	32-pad, 5 x 5 x 1.0 body, Lead Pitch 0.50mm Quad Flat No-Lead/Micro Lead Frame Package (QFN/MLF)
28P3	28-lead, 0.300" Wide, Plastic Dual Inline Package (PDIP)

9.4 ATmega88PA

Speed (MHz) ⁽³⁾	Power Supply (V)	Ordering Code ⁽²⁾	Package ⁽¹⁾	Operational Range
20	1.8 - 5.5	ATmega88PA-AU	32A	Industrial (-40°C to 85°C)
		ATmega88PA-AUR ⁽⁵⁾	32A	
		ATmega88PA-CCU	32CC1	
		ATmega88PA-CCUR ⁽⁵⁾	32CC1	
		ATmega88PA-MMH ⁽⁴⁾	28M1	
		ATmega88PA-MMHR ⁽⁴⁾⁽⁵⁾	28M1	
		ATmega88PA-MU	32M1-A	
		ATmega88PA-MUR ⁽⁵⁾	32M1-A	
		ATmega88PA-PU	28P3	Industrial (-40°C to 105°C)
		ATmega88PA-AN	32A	
		ATmega88PA-ANR ⁽⁵⁾	32A	
		ATmega88PA-MMN	28M1	
		ATmega88PA-MMNR ⁽⁵⁾	28M1	
		ATmega88PA-MN	32M1-A	
ATmega88PA-MNR ⁽⁵⁾	32M1-A			
ATmega88PA-PN	28P3			

- Note:
1. This device can also be supplied in wafer form. Please contact your local Atmel sales office for detailed ordering information and minimum quantities.
 2. Pb-free packaging complies to the European Directive for Restriction of Hazardous Substances (RoHS directive). Also Halide free and fully Green.
 3. See ["Speed Grades" on page 322](#).
 4. NiPdAu Lead Finish.
 5. Tape & Reel.

Package Type	
32A	32-lead, Thin (1.0mm) Plastic Quad Flat Package (TQFP)
32CC1	32-ball, 4 x 4 x 0.6mm package, ball pitch 0.5 mm, Ultra Thin, Fine-Pitch Ball Grill Array (UFBGA)
28M1	28-pad, 4 x 4 x 1.0 body, Lead Pitch 0.45 mm Quad Flat No-Lead/Micro Lead Frame Package (QFN/MLF)
32M1-A	32-pad, 5 x 5 x 1.0 body, Lead Pitch 0.50 mm Quad Flat No-Lead/Micro Lead Frame Package (QFN/MLF)
28P3	28-lead, 0.300" Wide, Plastic Dual Inline Package (PDIP)

9.5 ATmega168A

Speed (MHz) ⁽³⁾	Power Supply (V)	Ordering Code ⁽²⁾	Package ⁽¹⁾	Operational Range
20	1.8 - 5.5	ATmega168A-AU ATmega168A-AUR ⁽⁵⁾ ATmega168A-CCU ATmega168A-CCUR ⁽⁵⁾ ATmega168A-MMH ⁽⁴⁾ ATmega168A-MMHR ⁽⁴⁾⁽⁵⁾ ATmega168A-MU ATmega168A-MUR ⁽⁵⁾ ATmega168A-PU	32A 32A 32CC1 32CC1 28M1 28M1 32M1-A 32M1-A 28P3	Industrial (-40°C to 85°C)

- Note:
1. This device can also be supplied in wafer form. Please contact your local Atmel sales office for detailed ordering information and minimum quantities.
 2. Pb-free packaging complies to the European Directive for Restriction of Hazardous Substances (RoHS directive). Also Halide free and fully Green.
 3. See "[Speed Grades](#)" on page 322
 4. NiPdAu Lead Finish.
 5. Tape & Reel.

Package Type	
32A	32-lead, Thin (1.0mm) Plastic Quad Flat Package (TQFP)
32CC1	32-ball, 4 x 4 x 0.6 mm package, ball pitch 0.5mm, Ultra Thin, Fine-Pitch Ball Grill Array (UFBGA)
28M1	28-pad, 4 x 4 x 1.0 body, Lead Pitch 0.45mm Quad Flat No-Lead/Micro Lead Frame Package (QFN/MLF)
32M1-A	32-pad, 5 x 5 x 1.0 body, Lead Pitch 0.50mm Quad Flat No-Lead/Micro Lead Frame Package (QFN/MLF)
28P3	28-lead, 0.300" Wide, Plastic Dual Inline Package (PDIP)

9.6 ATmega168PA

Speed (MHz) ⁽³⁾	Power Supply (V)	Ordering Code ⁽²⁾	Package ⁽¹⁾	Operational Range
20	1.8 - 5.5	ATmega168PA-AU ATmega168PA-AUR ⁽⁵⁾ ATmega168PA-CCU ATmega168PA-CCUR ⁽⁵⁾ ATmega168PA-MMH ⁽⁴⁾ ATmega168PA-MMHR ⁽⁴⁾⁽⁵⁾ ATmega168PA-MU ATmega168PA-MUR ⁽⁵⁾ ATmega168PA-PU	32A 32A 32CC1 32CC1 28M1 28M1 32M1-A 32M1-A 28P3	Industrial (-40°C to 85°C)
20	1.8 - 5.5	ATmega168PA-AN ATmega168PA-ANR ⁽⁵⁾ ATmega168PA-MN ATmega168PA-MNR ⁽⁵⁾ ATmega168PA-PN	32A 32A 32M1-A 32M1-A 28P3	Industrial (-40°C to 105°C)

- Note:
1. This device can also be supplied in wafer form. Please contact your local Atmel sales office for detailed ordering information and minimum quantities.
 2. Pb-free packaging complies to the European Directive for Restriction of Hazardous Substances (RoHS directive). Also Halide free and fully Green.
 3. See "[Speed Grades](#)" on page 322.
 4. NiPdAu Lead Finish.
 5. Tape & Reel.

Package Type	
32A	32-lead, Thin (1.0mm) Plastic Quad Flat Package (TQFP)
32CC1	32-ball, 4 x 4 x 0.6mm package, ball pitch 0.5mm, Ultra Thin, Fine-Pitch Ball Grill Array (UFBGA)
28M1	28-pad, 4 x 4 x 1.0 body, Lead Pitch 0.45mm Quad Flat No-Lead/Micro Lead Frame Package (QFN/MLF)
32M1-A	32-pad, 5 x 5 x 1.0 body, Lead Pitch 0.50mm Quad Flat No-Lead/Micro Lead Frame Package (QFN/MLF)
28P3	28-lead, 0.300" Wide, Plastic Dual Inline Package (PDIP)

9.7 ATmega328

Speed (MHz)	Power Supply (V)	Ordering Code ⁽²⁾	Package ⁽¹⁾	Operational Range
20 ⁽³⁾	1.8 - 5.5	ATmega328-AU ATmega328-AUR ⁽⁵⁾ ATmega328-MMH ⁽⁴⁾ ATmega328-MMHR ⁽⁴⁾⁽⁵⁾ ATmega328-MU ATmega328-MUR ⁽⁵⁾ ATmega328-PU	32A 32A 28M1 28M1 32M1-A 32M1-A 28P3	Industrial (-40°C to 85°C)

- Note:
1. This device can also be supplied in wafer form. Please contact your local Atmel sales office for detailed ordering information and minimum quantities.
 2. Pb-free packaging complies to the European Directive for Restriction of Hazardous Substances (RoHS directive). Also Halide free and fully Green.
 3. See [Figure 29-1 on page 322](#).
 4. NiPdAu Lead Finish.
 5. Tape & Reel

Package Type	
32A	32-lead, Thin (1.0mm) Plastic Quad Flat Package (TQFP)
28M1	28-pad, 4 x 4 x 1.0 body, Lead Pitch 0.45mm Quad Flat No-Lead/Micro Lead Frame Package (QFN/MLF)
28P3	28-lead, 0.300" Wide, Plastic Dual Inline Package (PDIP)
32M1-A	32-pad, 5 x 5 x 1.0 body, Lead Pitch 0.50mm Quad Flat No-Lead/Micro Lead Frame Package (QFN/MLF)

9.8 ATmega328P

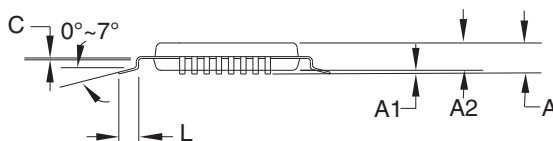
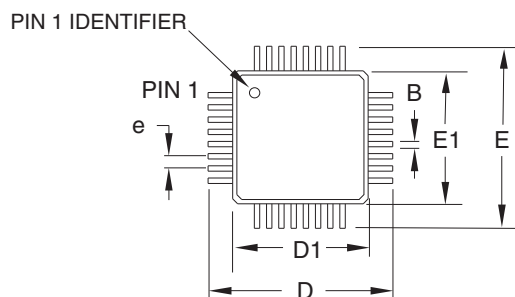
Speed (MHz) ⁽³⁾	Power Supply (V)	Ordering Code ⁽²⁾	Package ⁽¹⁾	Operational Range
20	1.8 - 5.5	ATmega328P-AU	32A	Industrial (-40°C to 85°C)
		ATmega328P-AUR ⁽⁵⁾	32A	
		ATmega328P-MMH ⁽⁴⁾	28M1	
		ATmega328P-MMHR ⁽⁴⁾⁽⁵⁾	28M1	
		ATmega328P-MU	32M1-A	
		ATmega328P-MUR ⁽⁵⁾	32M1-A	
		ATmega328P-PU	28P3	Industrial (-40°C to 105°C)
		ATmega328P-AN	32A	
		ATmega328P-ANR ⁽⁵⁾	32A	
		ATmega328P-MN	32M1-A	
ATmega328P-MNR ⁽⁵⁾	32M1-A			
ATmega328P-PN	28P3			

- Note:
1. This device can also be supplied in wafer form. Please contact your local Atmel sales office for detailed ordering information and minimum quantities.
 2. Pb-free packaging complies to the European Directive for Restriction of Hazardous Substances (RoHS directive). Also Halide free and fully Green.
 3. See [Figure 29-1 on page 322](#).
 4. NiPdAu Lead Finish.
 5. Tape & Reel.

Package Type	
32A	32-lead, Thin (1.0mm) Plastic Quad Flat Package (TQFP)
28M1	28-pad, 4 x 4 x 1.0 body, Lead Pitch 0.45mm Quad Flat No-Lead/Micro Lead Frame Package (QFN/MLF)
28P3	28-lead, 0.300" Wide, Plastic Dual Inline Package (PDIP)
32M1-A	32-pad, 5 x 5 x 1.0 body, Lead Pitch 0.50mm Quad Flat No-Lead/Micro Lead Frame Package (QFN/MLF)

10. Packaging Information

10.1 32A



COMMON DIMENSIONS
(Unit of Measure = mm)

SYMBOL	MIN	NOM	MAX	NOTE
A	–	–	1.20	
A1	0.05	–	0.15	
A2	0.95	1.00	1.05	
D	8.75	9.00	9.25	
D1	6.90	7.00	7.10	Note 2
E	8.75	9.00	9.25	
E1	6.90	7.00	7.10	Note 2
B	0.30	–	0.45	
C	0.09	–	0.20	
L	0.45	–	0.75	
e	0.80 TYP			

Notes:

1. This package conforms to JEDEC reference MS-026, Variation ABA.
2. Dimensions D1 and E1 do not include mold protrusion. Allowable protrusion is 0.25 mm per side. Dimensions D1 and E1 are maximum plastic body size dimensions including mold mismatch.
3. Lead coplanarity is 0.10 mm maximum.

2010-10-20



2325 Orchard Parkway
San Jose, CA 95131

TITLE

32A, 32-lead, 7 x 7 mm Body Size, 1.0 mm Body Thickness,
0.8 mm Lead Pitch, Thin Profile Plastic Quad Flat Package (TQFP)

DRAWING NO.

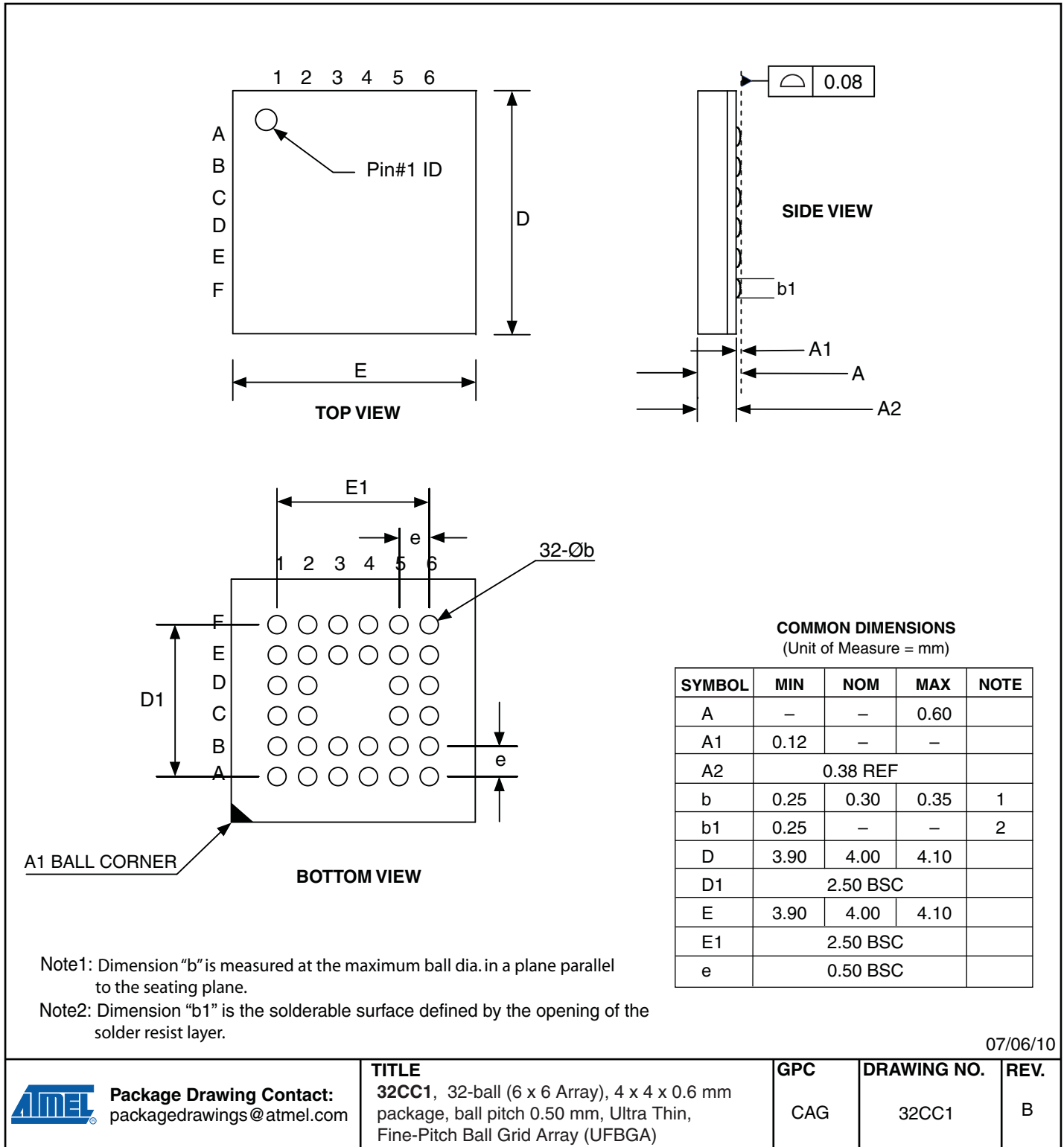
32A

REV.

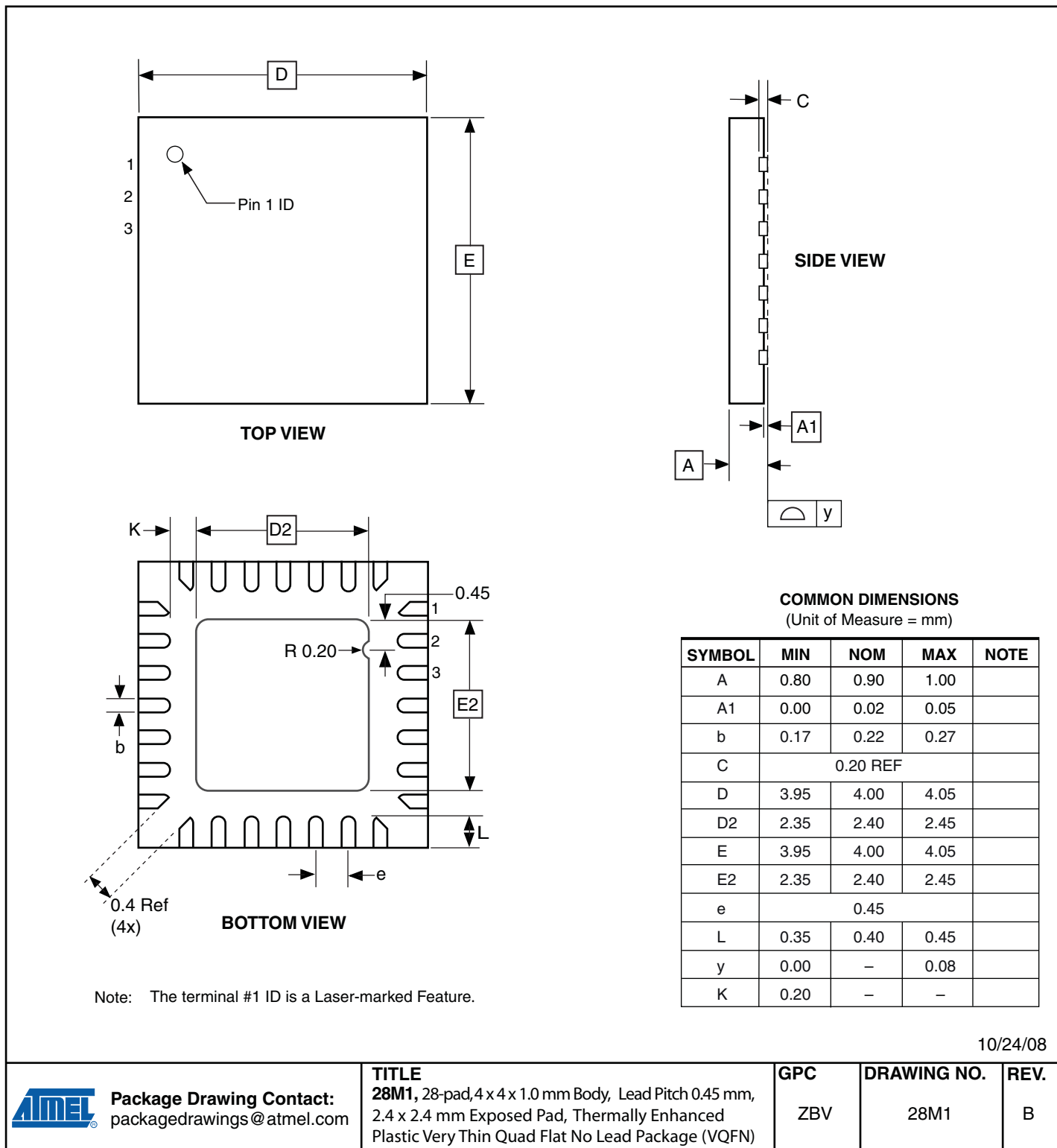
C



10.2 32CC1



10.3 28M1



10/24/08



Package Drawing Contact:
packagedrawings@atmel.com

TITLE
28M1, 28-pad, 4 x 4 x 1.0 mm Body, Lead Pitch 0.45 mm,
2.4 x 2.4 mm Exposed Pad, Thermally Enhanced
Plastic Very Thin Quad Flat No Lead Package (VQFN)

GPC

ZBV

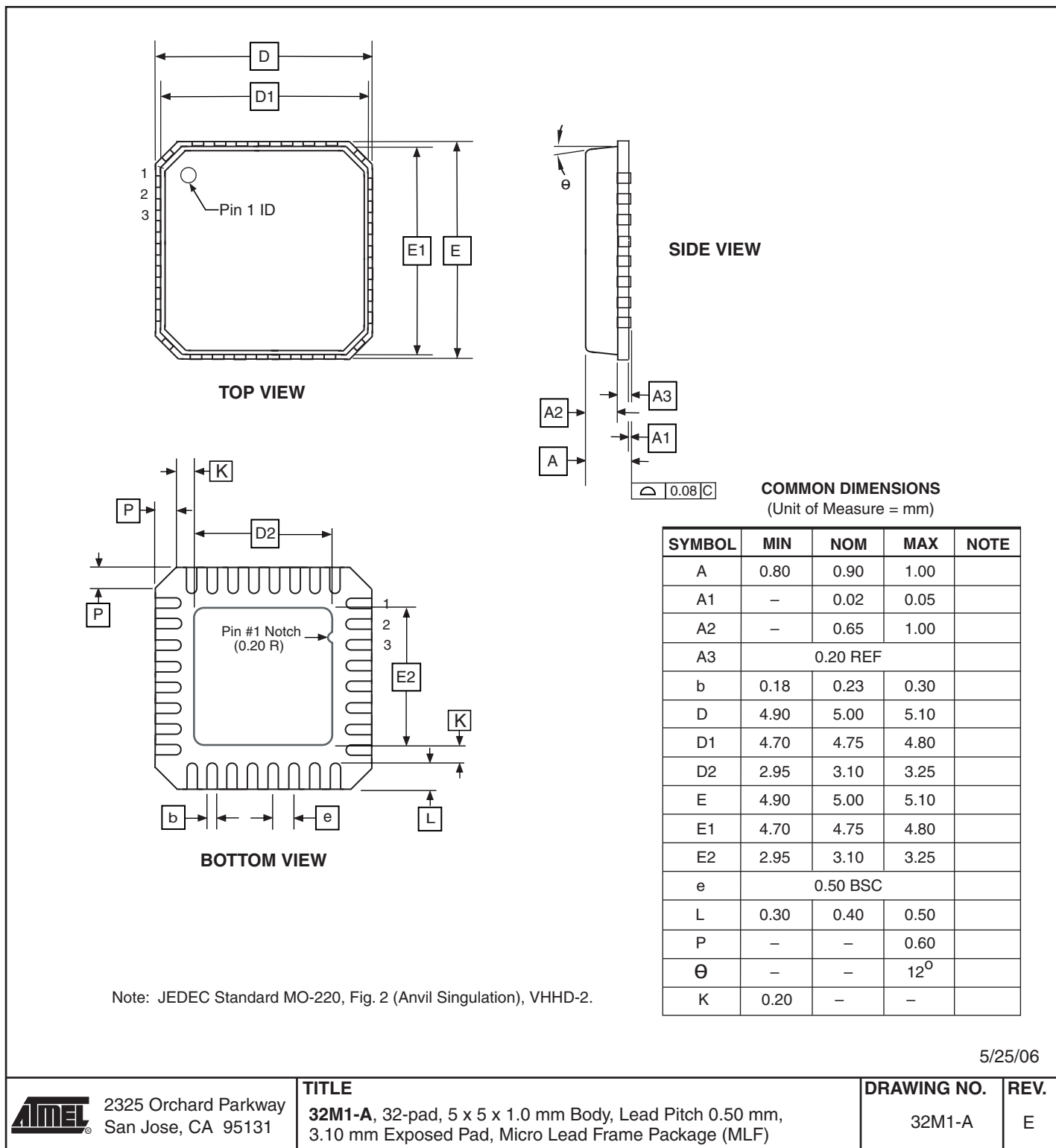
DRAWING NO.

28M1

REV.

B

10.4 32M1-A



5/25/06

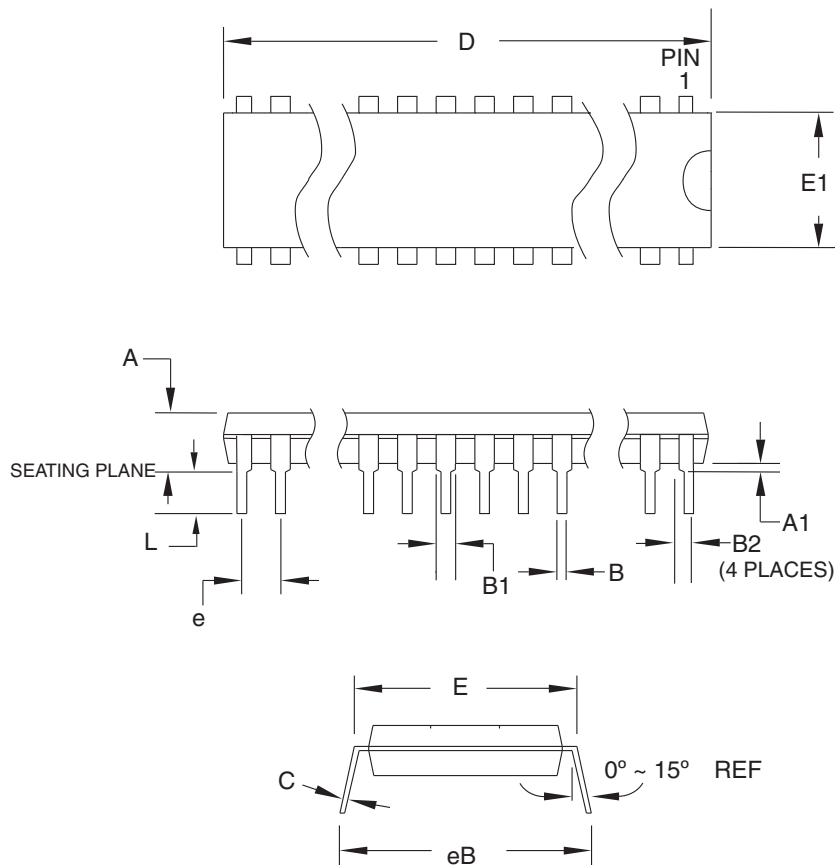
ATMEL 2325 Orchard Parkway
San Jose, CA 95131

TITLE
32M1-A, 32-pad, 5 x 5 x 1.0 mm Body, Lead Pitch 0.50 mm,
3.10 mm Exposed Pad, Micro Lead Frame Package (MLF)

DRAWING NO.
32M1-A

REV.
E

10.5 28P3



COMMON DIMENSIONS
(Unit of Measure = mm)

SYMBOL	MIN	NOM	MAX	NOTE
A	–	–	4.5724	
A1	0.508	–	–	
D	34.544	–	34.798	Note 1
E	7.620	–	8.255	
E1	7.112	–	7.493	Note 1
B	0.381	–	0.533	
B1	1.143	–	1.397	
B2	0.762	–	1.143	
L	3.175	–	3.429	
C	0.203	–	0.356	
eB	–	–	10.160	
e	2.540 TYP			

Note: 1. Dimensions D and E1 do not include mold Flash or Protrusion.
Mold Flash or Protrusion shall not exceed 0.25 mm (0.010").

09/28/01



2325 Orchard Parkway
San Jose, CA 95131

TITLE

28P3, 28-lead (0.300"/7.62 mm Wide) Plastic Dual
Inline Package (PDIP)

DRAWING NO.

28P3

REV.

B



11. Errata

11.1 Errata ATmega48A

The revision letter in this section refers to the revision of the ATmega48A device.

11.1.1 Rev. D

- **Analog MUX can be turned off when setting ACME bit**

1. **Analog MUX can be turned off when setting ACME bit**

If the ACME (Analog Comparator Multiplexer Enabled) bit in ADCSR is set while MUX3 in ADMUX is '1' (ADMUX[3:0]=1xxx), all MUX'es are turned off until the ACME bit is cleared.

Problem Fix/Workaround

Clear the MUX3 bit before setting the ACME bit.

11.2 Errata ATmega48PA

The revision letter in this section refers to the revision of the ATmega48PA device.

11.2.1 Rev. D

- **Analog MUX can be turned off when setting ACME bit**

1. **Analog MUX can be turned off when setting ACME bit**

If the ACME (Analog Comparator Multiplexer Enabled) bit in ADCSR is set while MUX3 in ADMUX is '1' (ADMUX[3:0]=1xxx), all MUX'es are turned off until the ACME bit is cleared.

Problem Fix/Workaround

Clear the MUX3 bit before setting the ACME bit.

11.3 Errata ATmega88A

The revision letter in this section refers to the revision of the ATmega88A device.

11.3.1 Rev. F

- **Analog MUX can be turned off when setting ACME bit**

1. **Analog MUX can be turned off when setting ACME bit**

If the ACME (Analog Comparator Multiplexer Enabled) bit in ADCSR is set while MUX3 in ADMUX is '1' (ADMUX[3:0]=1xxx), all MUX'es are turned off until the ACME bit is cleared.

Problem Fix/Workaround

Clear the MUX3 bit before setting the ACME bit.

11.4 Errata ATmega88PA

The revision letter in this section refers to the revision of the ATmega88PA device.

11.4.1 Rev. F

- **Analog MUX can be turned off when setting ACME bit**

1. **Analog MUX can be turned off when setting ACME bit**

If the ACME (Analog Comparator Multiplexer Enabled) bit in ADCSRB is set while MUX3 in ADMUX is '1' (ADMUX[3:0]=1xxx), all MUX'es are turned off until the ACME bit is cleared.

Problem Fix/Workaround

Clear the MUX3 bit before setting the ACME bit.

11.5 Errata ATmega168A

The revision letter in this section refers to the revision of the ATmega168A device.

11.5.1 Rev. E

- **Analog MUX can be turned off when setting ACME bit**

1. **Analog MUX can be turned off when setting ACME bit**

If the ACME (Analog Comparator Multiplexer Enabled) bit in ADCSRB is set while MUX3 in ADMUX is '1' (ADMUX[3:0]=1xxx), all MUX'es are turned off until the ACME bit is cleared.

Problem Fix/Workaround

Clear the MUX3 bit before setting the ACME bit.

11.6 Errata ATmega168PA

The revision letter in this section refers to the revision of the ATmega168PA device.

11.6.1 Rev E

- **Analog MUX can be turned off when setting ACME bit**

1. **Analog MUX can be turned off when setting ACME bit**

If the ACME (Analog Comparator Multiplexer Enabled) bit in ADCSRB is set while MUX3 in ADMUX is '1' (ADMUX[3:0]=1xxx), all MUX'es are turned off until the ACME bit is cleared.

Problem Fix/Workaround

Clear the MUX3 bit before setting the ACME bit.

11.7 Errata ATmega328

The revision letter in this section refers to the revision of the ATmega328 device.

11.7.1 Rev D

- **Analog MUX can be turned off when setting ACME bit**

1. **Analog MUX can be turned off when setting ACME bit**

If the ACME (Analog Comparator Multiplexer Enabled) bit in ADCSRB is set while MUX3 in ADMUX is '1' (ADMUX[3:0]=1xxx), all MUX'es are turned off until the ACME bit is cleared.

Problem Fix/Workaround

Clear the MUX3 bit before setting the ACME bit.

11.7.2 Rev C

Not sampled.

11.7.3 Rev B

- **Analog MUX can be turned off when setting ACME bit**
- **Unstable 32kHz Oscillator**

1. **Analog MUX can be turned off when setting ACME bit**

If the ACME (Analog Comparator Multiplexer Enabled) bit in ADCSRB is set while MUX3 in ADMUX is '1' (ADMUX[3:0]=1xxx), all MUX'es are turned off until the ACME bit is cleared.

Problem Fix/Workaround

Clear the MUX3 bit before setting the ACME bit.

2. **Unstable 32kHz Oscillator**

The 32kHz oscillator does not work as system clock. The 32kHz oscillator used as asynchronous timer is inaccurate.

Problem Fix/ Workaround

None.

11.7.4 Rev A

- **Analog MUX can be turned off when setting ACME bit**
- **Unstable 32kHz Oscillator**

1. **Analog MUX can be turned off when setting ACME bit**

If the ACME (Analog Comparator Multiplexer Enabled) bit in ADCSRB is set while MUX3 in ADMUX is '1' (ADMUX[3:0]=1xxx), all MUX'es are turned off until the ACME bit is cleared.

Problem Fix/Workaround

Clear the MUX3 bit before setting the ACME bit.

2. **Unstable 32kHz Oscillator**

The 32kHz oscillator does not work as system clock. The 32kHz oscillator used as asynchronous timer is inaccurate.

Problem Fix/ Workaround

None.

11.8 Errata ATmega328P

The revision letter in this section refers to the revision of the ATmega328P device.

11.8.1 Rev D

- **Analog MUX can be turned off when setting ACME bit**

1. **Analog MUX can be turned off when setting ACME bit**

If the ACME (Analog Comparator Multiplexer Enabled) bit in ADCSRB is set while MUX3 in ADMUX is '1' (ADMUX[3:0]=1xxx), all MUX'es are turned off until the ACME bit is cleared.

Problem Fix/Workaround

Clear the MUX3 bit before setting the ACME bit.

11.8.2 Rev C

Not sampled.

11.8.3 Rev B

- **Analog MUX can be turned off when setting ACME bit**
- **Unstable 32kHz Oscillator**

1. **Analog MUX can be turned off when setting ACME bit**

If the ACME (Analog Comparator Multiplexer Enabled) bit in ADCSRB is set while MUX3 in ADMUX is '1' (ADMUX[3:0]=1xxx), all MUX'es are turned off until the ACME bit is cleared.

Problem Fix/Workaround

Clear the MUX3 bit before setting the ACME bit.

2. **Unstable 32kHz Oscillator**

The 32kHz oscillator does not work as system clock. The 32kHz oscillator used as asynchronous timer is inaccurate.

Problem Fix/ Workaround

None.

11.8.4 Rev A

- **Unstable 32kHz Oscillator**

1. **Unstable 32kHz Oscillator**

The 32kHz oscillator does not work as system clock. The 32kHz oscillator used as asynchronous timer is inaccurate.

Problem Fix/ Workaround

None.

12. Datasheet Revision History

Please note that the referring page numbers in this section are referred to this document. The referring revision in this section are referring to the document revision.

12.1 Rev. 8271D – 05/11

1. Added Atmel QTouch Sensing Capability Feature
2. Updated "Register Description" on page 94 with PINxn as R/W.
3. Added a footnote to the PINxn, page 94.
4. Updated
5. Updated "Ordering Information", "ATmega328" on page 546. Added "ATmega328-MMH" and "ATmega328-MMHR".
6. Updated "Ordering Information", "ATmega328P" on page 547. Added "ATmega328P-MMH" and "ATmega328P-MMHR".
7. Added "Ordering Information" for ATmega48PA/88PA/168PA/328P @ 105°C
8. Updated "Errata ATmega328" on page 555 and "Errata ATmega328P" on page 556
98. Updated the datasheet according to the Atmel new brand style guide.

12.2 Rev. 8271C – 08/10

1. Added 32UFBGA Pinout, Table 1-1 on page 2.
2. Updated the "SRAM Data Memory", Figure 8-3 on page 19.
3. Updated "Ordering Information" on page 540 with CCU and CCUR code related to "32CC1" Package drawing.
4. "32CC1" Package drawing added on "Packaging Information" on page 548.

12.3 Rev. 8271B – 04/10

1. Updated Table 9-8 with correct value for timer oscillator at xtal2/tos2
2. Corrected use of SBIS instructions in assembly code examples.
3. Corrected BOD and BODSE bits to R/W in Section 10.11.2 on page 46, Section 12.5 on page 70 and Section 14.4 on page 94
4. Figures for bandgap characterization added, Figure 30-34 on page 350, Figure 30-81 on page 375, Figure 30-128 on page 400, Figure 30-175 on page 425, Figure 30-222 on page 450, Figure 30-269 on page 475, Figure 30-316 on page 500 and Figure 30-363 on page 525.
5. Updated "Packaging Information" on page 548 by replacing 28M1 with a correct corresponding package.

12.4 Rev. 8271A – 12/09

1. New datasheet 8271 with merged information for ATmega48PA, ATmega88PA, ATmega168PA and ATmega48A, ATmega88A and ATmega168A. Also included information on ATmega328 and ATmega328P
2. Changes done:
 - New devices added: ATmega48A/ATmega88A/ATmega168A and ATmega328
 - Updated Feature Description
 - Updated [Table 2-1 on page 6](#)
 - Added note for BOD Disable on [page 41](#).
 - Added note on BOD and BODSE in "MCUCR – MCU Control Register" on [page 94](#) and "Register Description" on [page 295](#)
 - Added limitation information for the application "Boot Loader Support – Read-While-Write Self-Programming" on [page 280](#)
 - Added limitation information for "Program And Data Memory Lock Bits" on [page 297](#)
 - Added specified DC characteristics
 - Added typical characteristics
 - Removed exception information in "Address Match Unit" on [page 224](#).



Atmel Corporation

2325 Orchard Parkway
San Jose, CA 95131
USA

Tel: (+1)(408) 441-0311

Fax: (+1)(408) 487-2600

www.atmel.com

Atmel Asia Limited

Unit 1-5 & 16, 19/F
BEA Tower, Millennium City 5
418 Kwun Tong Road
Kwun Tong, Kowloon

HONG KONG

Tel: (+852) 2245-6100

Fax: (+852) 2722-1369

Atmel Munich GmbH

Business Campus
Parkring 4
D-85748 Garching b. Munich
GERMANY

Tel: (+49) 89-31970-0

Fax: (+49) 89-3194621

Atmel Japan

9F, Tonetsu Shinkawa Bldg.
1-24-8 Shinkawa
Chuo-ku, Tokyo 104-0033
JAPAN

Tel: (+81)(3) 3523-3551

Fax: (+81)(3) 3523-7581

© 2011 Atmel Corporation. All rights reserved.

Atmel[®], Atmel logo and combinations thereof, AVR[®] and others are registered trademarks or trademarks of Atmel Corporation or its subsidiaries. Other terms and product names may be trademarks of others.

Disclaimer: The information in this document is provided in connection with Atmel products. No license, express or implied, by estoppel or otherwise, to any intellectual property right is granted by this document or in connection with the sale of Atmel products. **EXCEPT AS SET FORTH IN THE ATMEL TERMS AND CONDITIONS OF SALES LOCATED ON THE ATMEL WEBSITE, ATMEL ASSUMES NO LIABILITY WHATSOEVER AND DISCLAIMS ANY EXPRESS, IMPLIED OR STATUTORY WARRANTY RELATING TO ITS PRODUCTS INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. IN NO EVENT SHALL ATMEL BE LIABLE FOR ANY DIRECT, INDIRECT, CONSEQUENTIAL, PUNITIVE, SPECIAL OR INCIDENTAL DAMAGES (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS AND PROFITS, BUSINESS INTERRUPTION, OR LOSS OF INFORMATION) ARISING OUT OF THE USE OR INABILITY TO USE THIS DOCUMENT, EVEN IF ATMEL HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.** Atmel makes no representations or warranties with respect to the accuracy or completeness of the contents of this document and reserves the right to make changes to specifications and product descriptions at any time without notice. Atmel does not make any commitment to update the information contained herein. Unless specifically provided otherwise, Atmel products are not suitable for, and shall not be used in, automotive applications. Atmel products are not intended, authorized, or warranted for use as components in applications intended to support or sustain life.

ANEXO 5. HOJA DE ESPECIFICACIONES TÉCNICAS PROCESADOR
ATMEGA32U4

Features

- High Performance, Low Power AVR® 8-Bit Microcontroller
- Advanced RISC Architecture
 - 135 Powerful Instructions – Most Single Clock Cycle Execution
 - 32 x 8 General Purpose Working Registers
 - Fully Static Operation
 - Up to 16 MIPS Throughput at 16 MHz
 - On-Chip 2-cycle Multiplier
- Non-volatile Program and Data Memories
 - 16/32K Bytes of In-System Self-Programmable Flash
 - Endurance: 100,000 Write/Erase Cycles
 - Optional Boot Code Section with Independent Lock Bits
 - In-System Programming by On-chip Boot Program hardware activated after reset
 - True Read-While-Write Operation
 - All supplied parts are preprogrammed with a default USB bootloader
 - 1.25/2.5K Bytes Internal SRAM
 - 512Bytes/1K Bytes Internal EEPROM
 - Endurance: 100,000 Write/Erase Cycles
 - Programming Lock for Software Security
- JTAG (IEEE std. 1149.1 compliant) Interface
 - Boundary-scan Capabilities According to the JTAG Standard
 - Extensive On-chip Debug Support
 - Programming of Flash, EEPROM, Fuses, and Lock Bits through the JTAG Interface
- USB 2.0 Full-speed/Low Speed Device Module with Interrupt on Transfer Completion
 - Complies fully with Universal Serial Bus Specification Rev 2.0
 - Supports data transfer rates up to 12 Mbit/s and 1.5 Mbit/s
 - Endpoint 0 for Control Transfers: up to 64-bytes
 - 6 Programmable Endpoints with IN or Out Directions and with Bulk, Interrupt or Isochronous Transfers
 - Configurable Endpoints size up to 256 bytes in double bank mode
 - Fully independent 832 bytes USB DPRAM for endpoint memory allocation
 - Suspend/Resume Interrupts
 - CPU Reset possible on USB Bus Reset detection
 - 48 MHz from PLL for Full-speed Bus Operation
 - USB Bus Connection/Disconnection on Microcontroller Request
- Peripheral Features
 - On-chip PLL for USB and High Speed Timer: 32 up to 96 MHz operation
 - Two 8-bit Timer/Counters with Separate Prescaler and Compare Mode
 - Two 16-bit Timer/Counter with Separate Prescaler, Compare- and Capture Mode
 - One 10-bit High-Speed Timer/Counter with PLL (64 MHz) and Compare Mode
 - Four 8-bit PWM Channels
 - Four PWM Channels with Programmable Resolution from 2 to 16 Bits
 - Six PWM Channels for High Speed Operation, with Programmable Resolution from 2 to 11 Bits
 - Output Compare Modulator
 - 12-channels, 10-bit ADC (features Differential Channels with Programmable Gain)
 - Programmable Serial USART with Hardware Flow Control
 - Master/Slave SPI Serial Interface



**8-bit AVR®
Microcontroller
with
16/32K Bytes of
ISP Flash
and USB
Controller**

**ATmega16U4
ATmega32U4**

Preliminary

Summary

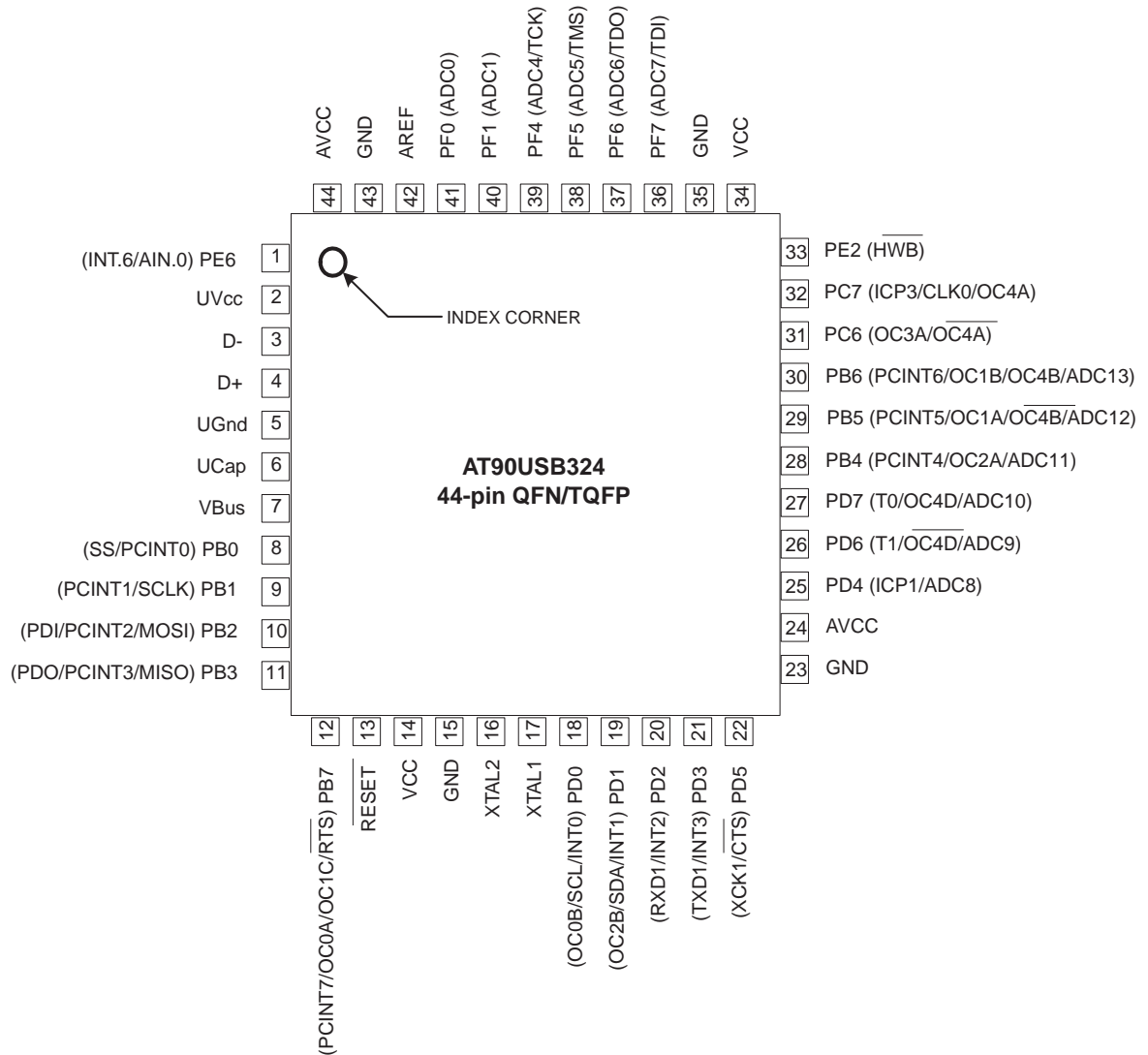
7766BS-AVR-07/08



- Byte Oriented 2-wire Serial Interface
- Programmable Watchdog Timer with Separate On-chip Oscillator
- On-chip Analog Comparator
- Interrupt and Wake-up on Pin Change (8xPCINT + 5xINT sources)
- On-chip Temperature Sensor (see A/D Converter section)
- Special Microcontroller Features
 - Power-on Reset and Programmable Brown-out Detection
 - Internal 8 MHz Calibrated Oscillator
 - Internal clock prescaler & On-the-fly Clock Switching (Int RC / Ext Osc)
 - External and Internal Interrupt Sources
 - Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby, and Extended Standby
- I/O and Packages
 - All I/O combine CMOS outputs and LVTTL inputs
 - 26 Programmable I/O Lines
 - 44-lead TQFP Package, 10x10mm
 - 44-lead QFN Package, 7x7mm
- Operating Voltages
 - 2.7 - 5.5V
- Operating temperature
 - Industrial (-40°C to +85°C)
- Maximum Frequency
 - 8 MHz at 2.7V - Industrial range
 - 16 MHz at 4.5V - Industrial range

1. Pin Configurations

Figure 1-1. Pinout ATmega16U4/ATmega32U4



1.1 Disclaimer

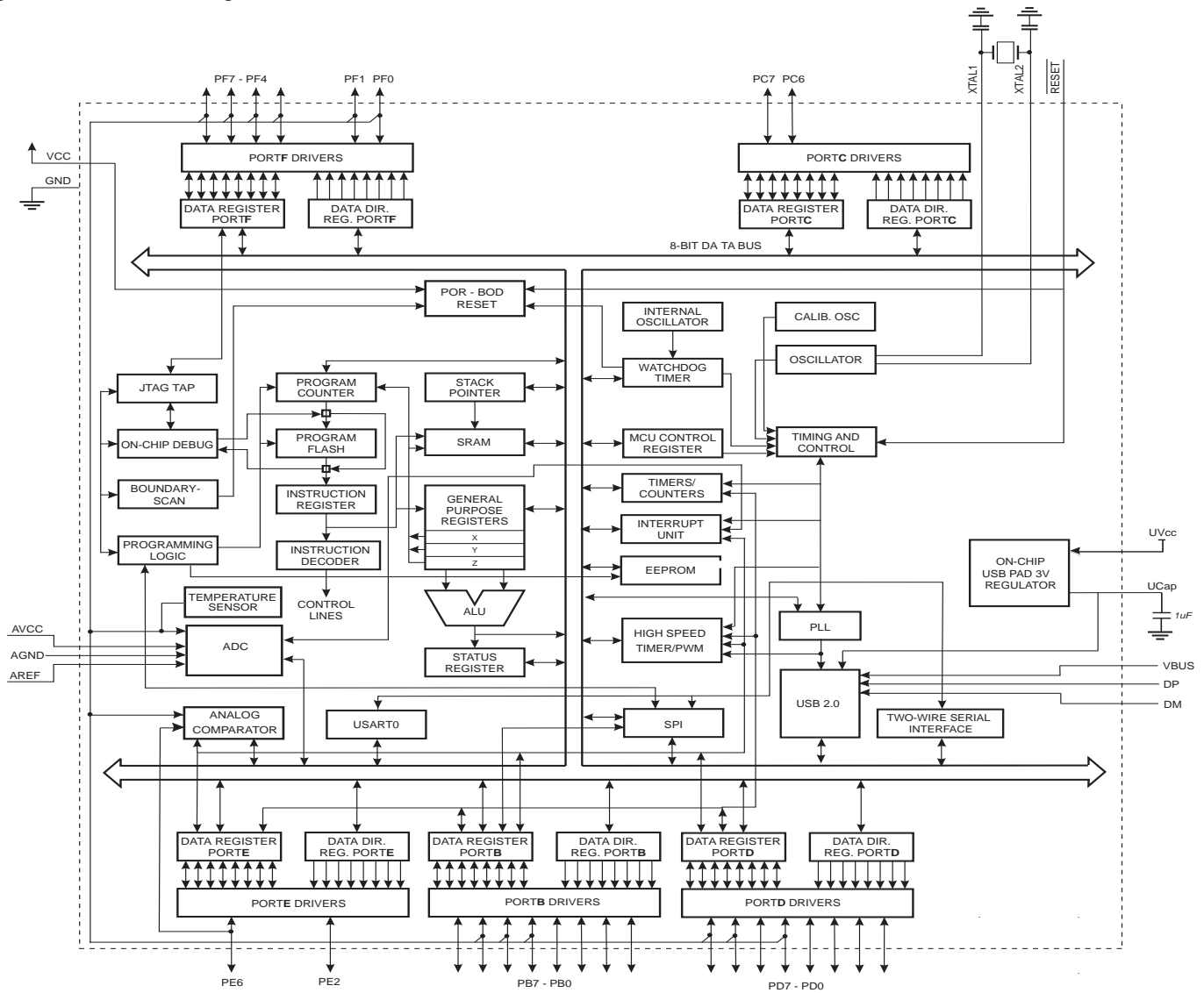
Typical values contained in this datasheet are based on simulations and characterization of other AVR microcontrollers manufactured on the same process technology. Min and Max values will be available after the device is characterized.

2. Overview

The ATmega16U4/ATmega32U4 is a low-power CMOS 8-bit microcontroller based on the AVR enhanced RISC architecture. By executing powerful instructions in a single clock cycle, the ATmega16U4/ATmega32U4 achieves throughputs approaching 1 MIPS per MHz allowing the system designer to optimize power consumption versus processing speed.

2.1 Block Diagram

Figure 2-1. Block Diagram



- Subject to changes -

The AVR core combines a rich instruction set with 32 general purpose working registers. All the 32 registers are directly connected to the Arithmetic Logic Unit (ALU), allowing two independent registers to be accessed in one single instruction executed in one clock cycle. The resulting architecture is more code efficient while achieving throughputs up to ten times faster than conventional CISC microcontrollers.

The ATmega16U4/ATmega32U4 provides the following features: 16/32K bytes of In-System Programmable Flash with Read-While-Write capabilities, 512Bytes/1K bytes EEPROM, 1.25/2.5K bytes SRAM, 26 general purpose I/O lines (CMOS outputs and LVTTTL inputs), 32 general purpose working registers, four flexible Timer/Counters with compare modes and PWM, one more high-speed Timer/Counter with compare modes and PLL adjustable source, one USART (including CTS/RTS flow control signals), a byte oriented 2-wire Serial Interface, a 12-

channels 10-bit ADC with optional differential input stage with programmable gain, an on-chip calibrated temperature sensor, a programmable Watchdog Timer with Internal Oscillator, an SPI serial port, IEEE std. 1149.1 compliant JTAG test interface, also used for accessing the On-chip Debug system and programming and six software selectable power saving modes. The Idle mode stops the CPU while allowing the SRAM, Timer/Counters, SPI port, and interrupt system to continue functioning. The Power-down mode saves the register contents but freezes the Oscillator, disabling all other chip functions until the next interrupt or Hardware Reset. The ADC Noise Reduction mode stops the CPU and all I/O modules except ADC, to minimize switching noise during ADC conversions. In Standby mode, the Crystal/Resonator Oscillator is running while the rest of the device is sleeping. This allows very fast start-up combined with low power consumption.

The device is manufactured using ATMEL's high-density nonvolatile memory technology. The On-chip ISP Flash allows the program memory to be reprogrammed in-system through an SPI serial interface, by a conventional nonvolatile memory programmer, or by an On-chip Boot program running on the AVR core. The boot program can use any interface to download the application program in the application Flash memory. Software in the Boot Flash section will continue to run while the Application Flash section is updated, providing true Read-While-Write operation. By combining an 8-bit RISC CPU with In-System Self-Programmable Flash on a monolithic chip, the ATMEL ATmega16U4/ATmega32U4 is a powerful microcontroller that provides a highly flexible and cost effective solution to many embedded control applications.

The ATmega16U4/ATmega32U4 AVR is supported with a full suite of program and system development tools including: C compilers, macro assemblers, program debugger/simulators, in-circuit emulators, and evaluation kits.

2.2 Pin Descriptions

2.2.1 VCC

Digital supply voltage.

2.2.2 GND

Ground.

2.2.3 Port B (PB7..PB0)

Port B is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port B output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port B pins that are externally pulled low will source current if the pull-up resistors are activated. The Port B pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Port B has better driving capabilities than the other ports.

Port B also serves the functions of various special features of the ATmega16U4/ATmega32U4 as listed on [page 70](#).

2.2.4 Port C (PC7,PC6)

Port C is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port C output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port C pins that are externally pulled low will source current if the pull-up resistors are activated. The Port C pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Only bits 6 and 7 are present on the product pinout.

Port C also serves the functions of special features of the ATmega16U4/ATmega32U4 as listed on [page 73](#).

2.2.5 Port D (PD7..PD0)

Port D is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port D output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port D pins that are externally pulled low will source current if the pull-up resistors are activated. The Port D pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Port D also serves the functions of various special features of the ATmega16U4/ATmega32U4 as listed on [page 75](#).

2.2.6 Port E (PE6,PE2)

Port E is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port E output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port E pins that are externally pulled low will source current if the pull-up resistors are activated. The Port E pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Only bits 2 and 6 are present on the product pinout.

Port E also serves the functions of various special features of the ATmega16U4/ATmega32U4 as listed on [page 78](#).

2.2.7 Port F (PF7..PF4, PF1,PF0)

Port F serves as analog inputs to the A/D Converter.

Port F also serves as an 8-bit bi-directional I/O port, if the A/D Converter channels are not used. Port pins can provide internal pull-up resistors (selected for each bit). The Port F output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port F pins that are externally pulled low will source current if the pull-up resistors are activated. The Port F pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Bits 2 and 3 are not present on the product pinout.

Port F also serves the functions of the JTAG interface. If the JTAG interface is enabled, the pull-up resistors on pins PF7(TDI), PF5(TMS), and PF4(TCK) will be activated even if a reset occurs.

2.2.8 D-

USB Full speed / Low Speed Negative Data Upstream Port. Should be connected to the USB D- connector pin with a serial 22 Ohms resistor.

2.2.9 D+

USB Full speed / Low Speed Positive Data Upstream Port. Should be connected to the USB D+ connector pin with a serial 22 Ohms resistor.

2.2.10 UGND

USB Pads Ground.

- 2.2.11 UVCC**
USB Pads Internal Regulator Input supply voltage.
- 2.2.12 UCAP**
USB Pads Internal Regulator Output supply voltage. Should be connected to an external capacitor (1 μ F).
- 2.2.13 VBUS**
USB VBUS monitor input.
- 2.2.14 $\overline{\text{RESET}}$**
Reset input. A low level on this pin for longer than the minimum pulse length will generate a reset, even if the clock is not running. The minimum pulse length is given in [Table 8-1 on page 48](#). Shorter pulses are not guaranteed to generate a reset.
- 2.2.15 XTAL1**
Input to the inverting Oscillator amplifier and input to the internal clock operating circuit.
- 2.2.16 XTAL2**
Output from the inverting Oscillator amplifier.
- 2.2.17 AVCC**
AVCC is the supply voltage pin (input) for all the A/D Converter channels. If the ADC is not used, it should be externally connected to V_{CC} . If the ADC is used, it should be connected to V_{CC} through a low-pass filter.
- 2.2.18 AREF**
This is the analog reference pin (input) for the A/D Converter.

3. About Code Examples

This documentation contains simple code examples that briefly show how to use various parts of the device. Be aware that not all C compiler vendors include bit definitions in the header files and interrupt handling in C is compiler dependent. Please confirm with the C compiler documentation for more details.

These code examples assume that the part specific header file is included before compilation. For I/O registers located in extended I/O map, "IN", "OUT", "SBIS", "SBIC", "CBI", and "SBI" instructions must be replaced with instructions that allow access to extended I/O. Typically "LDS" and "STS" combined with "SBRS", "SBRC", "SBR", and "CBR".



4. Register Summary

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Page		
(0xFF)	Reserved	-	-	-	-	-	-	-	-			
(0xFE)	Reserved	-	-	-	-	-	-	-	-			
(0xFD)	Reserved	-	-	-	-	-	-	-	-			
(0xFC)	Reserved	-	-	-	-	-	-	-	-			
(0xFB)	Reserved	-	-	-	-	-	-	-	-			
(0xFA)	Reserved	-	-	-	-	-	-	-	-			
(0xF9)	Reserved	-	-	-	-	-	-	-	-			
(0xF8)	Reserved	-	-	-	-	-	-	-	-			
(0xF7)	Reserved	-	-	-	-	-	-	-	-			
(0xF6)	Reserved	-	-	-	-	-	-	-	-			
(0xF5)	Reserved	-	-	-	-	-	-	-	-			
(0xF4)	UEINT	-	EPINT6:0						-	-		
(0xF3)	UEBCHX	-	-	-	-	-	BYCT10:8					
(0xF2)	UEBCLX	BYCT7:0										
(0xF1)	UEDATX	DAT7:0										
(0xF0)	UEIENX	FLERRE	NAKINE	-	NAKOUTE	RXSTPE	RXOUTE	STALLEDE	TXINE			
(0xEF)	UESTA1X	-	-	-	-	-	CTRLDIR	CURRBK1:0				
(0xEE)	UESTA0X	CFGOK	OVERFI	UNDERFI	-	DTSEQ1:0		NBUSYBK1:0				
(0xED)	UECFG1X	EPSIZE2:0		EPBK1:0		ALLOC	-	-	-			
(0xEC)	UECFG0X	EPTYPE1:0		-	-	-	-	-	EPDIR			
(0xEB)	UECONX	-	-	STALLRQ	STALLRQC	RSTDT	-	-	EPEN			
(0xEA)	UERST	-	EPRST6:0									
(0xE9)	UENUM	-	-	-	-	-	EPNUM2:0					
(0xE8)	UEINTX	FIFOCON	NAKINI	RWAL	NAKOUTI	RXSTPI	RXOUTI	STALLEDI	TXINI			
(0xE7)	Reserved	-	-	-	-	-	-	-	-			
(0xE6)	UDMFN	-	-	-	FNCERR	-	-	-	-			
(0xE5)	UDFNUMH	-	-	-	-	-	FNUM10:8					
(0xE4)	UDFNUML	FNUM7:0										
(0xE3)	UDADDR	ADDEN	UADD6:0							-	-	
(0xE2)	UDIEN	-	UPRSME	EORSME	WAKEUPE	EORSTE	SOFE	MSOFE	SUSPE			
(0xE1)	UDINT	-	UPRSMI	EORSMI	WAKEUPI	EORSTI	SOFI	MSOFI	SUSPI			
(0xE0)	UDCON	-	-	-	-	RSTCPU	LSM	RMWKUP	DETACH			
(0xDF)	Reserved	-	-	-	-	-	-	-	-			
(0xDE)	Reserved	-	-	-	-	-	-	-	-			
(0xDD)	Reserved	-	-	-	-	-	-	-	-			
(0xDC)	Reserved	-	-	-	-	-	-	-	-			
(0xDB)	Reserved	-	-	-	-	-	-	-	-			
(0xDA)	USBINT	-	-	-	-	-	-	-	VBUSTI			
(0xD9)	USBSTA	-	-	-	-	-	-	ID	VBUS			
(0xD8)	USBCON	USBE	-	FRZCLK	OTGPADE	-	-	-	VBUSTE			
(0xD7)	UHWCON	-	-	-	-	-	-	-	UVREGE			
(0xD6)	Reserved	-	-	-	-	-	-	-	-			
(0xD5)	Reserved	-	-	-	-	-	-	-	-			
(0xD4)	DT4	DT4H3	DT4H2	DT4H1	DT4H0	DT4L3	DT4L2	DT4L1	DT4L0			
(0xD3)	Reserved	-	-	-	-	-	-	-	-			
(0xD2)	OCR4D	Timer/Counter4 - Output Compare Register D										
(0xD1)	OCR4C	Timer/Counter4 - Output Compare Register C										
(0xD0)	OCR4B	Timer/Counter4 - Output Compare Register B										
(0xCF)	OCR4A	Timer/Counter4 - Output Compare Register A										
(0xCE)	UDR1	USART1 I/O Data Register										
(0xCD)	UBRR1H	-	-	-	-	USART1 Baud Rate Register High Byte						
(0xCC)	UBRR1L	USART1 Baud Rate Register Low Byte										
(0xCB)	Reserved	-	-	-	-	-	-	-	-			
(0xCA)	UCSR1C	UMSEL11	UMSEL10	UPM11	UPM10	USBS1	UCSZ11	UCSZ10	UCPOL1			
(0xC9)	UCSR1B	RXCIE1	TXCIE1	UDRIE1	RXEN1	TXEN1	UCSZ12	RXB81	TXB81			
(0xC8)	UCSR1A	RXC1	TXC1	UDRE1	FE1	DOR1	PE1	U2X1	MPCM1			
(0xC7)	CLKSTA	-	-	-	-	-	-	RCON	EXTON			
(0xC6)	CLKSEL1	RCCKSEL3	RCCKSEL2	RCCKSEL1	RCCKSEL0	EXCKSEL3	EXCKSEL2	EXCKSEL1	EXCKSEL0			
(0xC5)	CLKSEL0	RCSUT1	RCSUT0	EXSUT1	EXSUT0	RCE	EXTE	-	CLKS			
(0xC4)	TCCR4E	TLOCK4	ENHC4	OC4OE5	OC4OE4	OC4OE3	OC4OE2	OC4OE1	OC4OE0			
(0xC3)	TCCR4D	FPIE4	FPEN4	FPNC4	FPES4	FPAC4	FPF4	WGM41	WGM40			
(0xC2)	TCCR4C	COM4A1S	COM4A0S	COM4B1S	COM4B0S	COM4D1S	COM4D0S	FOC4D	PWM4D			
(0xC1)	TCCR4B	PWM4X	PSR4	DTPS41	DTPS40	CS43	CS42	CS41	CS40			
(0xC0)	TCCR4A	COM4A1	COM4A0	COM4B1	COM4B0	FOC4A	FOC4B	PWM4A	PWM4B			
(0xBF)	TC4H	-	-	-	-	-	Timer/Counter4 High Byte					

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Page
(0xBE)	TCNT4	Timer/Counter4 - Counter Register Low Byte								
(0xBD)	TWAMR	TWAM6	TWAM5	TWAM4	TWAM3	TWAM2	TWAM1	TWAM0	-	
(0xBC)	TWCR	TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	-	TWIE	
(0xBB)	TWDR	2-wire Serial Interface Data Register								
(0xBA)	TWAR	TWA6	TWA5	TWA4	TWA3	TWA2	TWA1	TWA0	TWGCE	
(0xB9)	TWSR	TWS7	TWS6	TWS5	TWS4	TWS3	-	TWPS1	TWPS0	
(0xB8)	TWBR	2-wire Serial Interface Bit Rate Register								
(0xB7)	Reserved	-	-	-	-	-	-	-	-	
(0xB6)	Reserved	-	-	-	-	-	-	-	-	
(0xB5)	Reserved	-	-	-	-	-	-	-	-	
(0xB4)	Reserved	-	-	-	-	-	-	-	-	
(0xB3)	Reserved	-	-	-	-	-	-	-	-	
(0xB2)	Reserved	-	-	-	-	-	-	-	-	
(0xB1)	Reserved	-	-	-	-	-	-	-	-	
(0xB0)	Reserved	-	-	-	-	-	-	-	-	
(0xAF)	Reserved	-	-	-	-	-	-	-	-	
(0xAE)	Reserved	-	-	-	-	-	-	-	-	
(0xAD)	Reserved	-	-	-	-	-	-	-	-	
(0xAC)	Reserved	-	-	-	-	-	-	-	-	
(0xAB)	Reserved	-	-	-	-	-	-	-	-	
(0xAA)	Reserved	-	-	-	-	-	-	-	-	
(0xA9)	Reserved	-	-	-	-	-	-	-	-	
(0xA8)	Reserved	-	-	-	-	-	-	-	-	
(0xA7)	Reserved	-	-	-	-	-	-	-	-	
(0xA6)	Reserved	-	-	-	-	-	-	-	-	
(0xA5)	Reserved	-	-	-	-	-	-	-	-	
(0xA4)	Reserved	-	-	-	-	-	-	-	-	
(0xA3)	Reserved	-	-	-	-	-	-	-	-	
(0xA2)	Reserved	-	-	-	-	-	-	-	-	
(0xA1)	Reserved	-	-	-	-	-	-	-	-	
(0xA0)	Reserved	-	-	-	-	-	-	-	-	
(0x9F)	Reserved	-	-	-	-	-	-	-	-	
(0x9E)	Reserved	-	-	-	-	-	-	-	-	
(0x9D)	OCR3CH	Timer/Counter3 - Output Compare Register C High Byte								
(0x9C)	OCR3CL	Timer/Counter3 - Output Compare Register C Low Byte								
(0x9B)	OCR3BH	Timer/Counter3 - Output Compare Register B High Byte								
(0x9A)	OCR3BL	Timer/Counter3 - Output Compare Register B Low Byte								
(0x99)	OCR3AH	Timer/Counter3 - Output Compare Register A High Byte								
(0x98)	OCR3AL	Timer/Counter3 - Output Compare Register A Low Byte								
(0x97)	ICR3H	Timer/Counter3 - Input Capture Register High Byte								
(0x96)	ICR3L	Timer/Counter3 - Input Capture Register Low Byte								
(0x95)	TCNT3H	Timer/Counter3 - Counter Register High Byte								
(0x94)	TCNT3L	Timer/Counter3 - Counter Register Low Byte								
(0x93)	Reserved	-	-	-	-	-	-	-	-	
(0x92)	TCCR3C	FOC3A	-	-	-	-	-	-	-	
(0x91)	TCCR3B	ICNC3	ICES3	-	WGM33	WGM32	CS32	CS31	CS30	
(0x90)	TCCR3A	COM3A1	COM3A0	COM3B1	COM3B0	COM3C1	COM3C0	WGM31	WGM30	
(0x8F)	Reserved	-	-	-	-	-	-	-	-	
(0x8E)	Reserved	-	-	-	-	-	-	-	-	
(0x8D)	OCR1CH	Timer/Counter1 - Output Compare Register C High Byte								
(0x8C)	OCR1CL	Timer/Counter1 - Output Compare Register C Low Byte								
(0x8B)	OCR1BH	Timer/Counter1 - Output Compare Register B High Byte								
(0x8A)	OCR1BL	Timer/Counter1 - Output Compare Register B Low Byte								
(0x89)	OCR1AH	Timer/Counter1 - Output Compare Register A High Byte								
(0x88)	OCR1AL	Timer/Counter1 - Output Compare Register A Low Byte								
(0x87)	ICR1H	Timer/Counter1 - Input Capture Register High Byte								
(0x86)	ICR1L	Timer/Counter1 - Input Capture Register Low Byte								
(0x85)	TCNT1H	Timer/Counter1 - Counter Register High Byte								
(0x84)	TCNT1L	Timer/Counter1 - Counter Register Low Byte								
(0x83)	Reserved	-	-	-	-	-	-	-	-	
(0x82)	TCCR1C	FOC1A	FOC1B	FOC1C	-	-	-	-	-	
(0x81)	TCCR1B	ICNC1	ICES1	-	WGM13	WGM12	CS12	CS11	CS10	
(0x80)	TCCR1A	COM1A1	COM1A0	COM1B1	COM1B0	COM1C1	COM1C0	WGM11	WGM10	
(0x7F)	DIDR1	-	-	-	-	-	-	-	AIN0D	
(0x7E)	DIDR0	ADC7D	ADC6D	ADC5D	ADC4D	-	-	ADC1D	ADC0D	
(0x7D)	DIDR2	-	-	ADC13D	ADC12D	ADC11D	ADC10D	ADC9D	ADC8D	

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Page
(0x7C)	ADMUX	REFS1	REFS0	ADLAR	MUX4	MUX3	MUX2	MUX1	MUX0	
(0x7B)	ADCSRB	ADHSM	ACME	MUX5	-	ADTS3	ADTS2	ADTS1	ADTS0	
(0x7A)	ADCSRA	ADEN	ADSC	ADATE	ADIF	ADIE	ADPS2	ADPS1	ADPS0	
(0x79)	ADCH	ADC Data Register High byte								
(0x78)	ADCL	ADC Data Register Low byte								
(0x77)	Reserved	-	-	-	-	-	-	-	-	
(0x76)	Reserved	-	-	-	-	-	-	-	-	
(0x75)	Reserved	-	-	-	-	-	-	-	-	
(0x74)	Reserved	-	-	-	-	-	-	-	-	
(0x73)	Reserved	-	-	-	-	-	-	-	-	
(0x72)	TIMSK4	OCIE4D	OCIE4A	OCIE4B	-	-	TOIE4	-	-	
(0x71)	TIMSK3	-	-	ICIE3	-	OCIE3C	OCIE3B	OCIE3A	TOIE3	
(0x70)	TIMSK2	-	-	-	-	-	OCIE2B	OCIE2A	TOIE2	
(0x6F)	TIMSK1	-	-	ICIE1	-	OCIE1C	OCIE1B	OCIE1A	TOIE1	
(0x6E)	TIMSK0	-	-	-	-	-	OCIE0B	OCIE0A	TOIE0	
(0x6D)	Reserved	-	-	-	-	-	-	-	-	
(0x6C)	Reserved	-	-	-	-	-	-	-	-	
(0x6B)	PCMSK0	PCINT7	PCINT6	PCINT5	PCINT4	PCINT3	PCINT2	PCINT1	PCINT0	
(0x6A)	EICRB	-	-	ISC61	ISC60	-	-	-	-	
(0x69)	EICRA	ISC31	ISC30	ISC21	ISC20	ISC11	ISC10	ISC01	ISC00	
(0x68)	PCICR	-	-	-	-	-	-	-	PCIE0	
(0x67)	RCCTRL	-	-	-	-	-	-	-	RCFREQ	
(0x66)	OSCCAL	RC Oscillator Calibration Register								
(0x65)	PRR1	PRUSB	-	-	PRTIM4	PRTIM3	-	-	PRUSART1	
(0x64)	PRR0	PRTWI	PRTIM2	PRTIM0	-	PRTIM1	PRSPI	-	PRADC	
(0x63)	Reserved	-	-	-	-	-	-	-	-	
(0x62)	Reserved	-	-	-	-	-	-	-	-	
(0x61)	CLKPR	CLKPCE	-	-	-	CLKPS3	CLKPS2	CLKPS1	CLKPS0	
(0x60)	WDTCR	WDIF	WDIE	WDP3	WDCE	WDE	WDP2	WDP1	WDP0	
0x3F (0x5F)	SREG	I	T	H	S	V	N	Z	C	
0x3E (0x5E)	SPH	SP15	SP14	SP13	SP12	SP11	SP10	SP9	SP8	
0x3D (0x5D)	SPL	SP7	SP6	SP5	SP4	SP3	SP2	SP1	SP0	
0x3C (0x5C)	Reserved	-	-	-	-	-	-	-	-	
0x3B (0x5B)	RAMPZ	-	-	-	-	-	-	RAMPZ1	RAMPZ0	
0x3A (0x5A)	Reserved	-	-	-	-	-	-	-	-	
0x39 (0x59)	Reserved	-	-	-	-	-	-	-	-	
0x38 (0x58)	Reserved	-	-	-	-	-	-	-	-	
0x37 (0x57)	SPMCSR	SPMIE	RWWSB	SIGRD	RWWSRE	BLBSET	PGWRT	PGERS	SPMEN	
0x36 (0x56)	Reserved	-	-	-	-	-	-	-	-	
0x35 (0x55)	MCUCR	JTD	-	-	PUD	-	-	IVSEL	IVCE	
0x34 (0x54)	MCUSR	-	-	USBRF	JTRF	WDRF	BORF	EXTRF	PORF	
0x33 (0x53)	SMCR	-	-	-	-	SM2	SM1	SM0	SE	
0x32 (0x52)	PLLFREQ	PINMUX	PLLUSB	PLLTM1	PLLTM0	PDIV3	PDIV2	PDIV1	PDIV0	
0x31 (0x51)	OCDR/ MONDR	OCDR7	OCDR6	OCDR5	OCDR4	OCDR3	OCDR2	OCDR1	OCDR0	
		Monitor Data Register								
0x30 (0x50)	ACSR	ACD	ACBG	ACO	ACI	ACIE	ACIC	ACIS1	ACIS0	
0x2F (0x4F)	Reserved	-	-	-	-	-	-	-	-	
0x2E (0x4E)	SPDR	SPI Data Register								
0x2D (0x4D)	SPSR	SPIF	WCOL	-	-	-	-	-	SPI2X	
0x2C (0x4C)	SPCR	SPIE	SPE	DORD	MSTR	CPOL	CPHA	SPR1	SPR0	
0x2B (0x4B)	GPIOR2	General Purpose I/O Register 2								
0x2A (0x4A)	GPIOR1	General Purpose I/O Register 1								
0x29 (0x49)	PLLCSR	-	-	-	PINDIV	-	-	PLLE	PLOCK	
0x28 (0x48)	OCR0B	Timer/Counter0 Output Compare Register B								
0x27 (0x47)	OCR0A	Timer/Counter0 Output Compare Register A								
0x26 (0x46)	TCNT0	Timer/Counter0 (8 Bit)								
0x25 (0x45)	TCCR0B	FOC0A	FOC0B	-	-	WGM02	CS02	CS01	CS00	
0x24 (0x44)	TCCR0A	COM0A1	COM0A0	COM0B1	COM0B0	-	-	WGM01	WGM00	
0x23 (0x43)	GTCCR	TSM	-	-	-	-	-	PSRASY	PSRSYNC	
0x22 (0x42)	EEARH	-	-	-	-	EEPROM Address Register High Byte				
0x21 (0x41)	EEARL	EEPROM Address Register Low Byte								
0x20 (0x40)	EEDR	EEPROM Data Register								
0x1F (0x3F)	EECR	-	-	EEMP1	EEMP0	EERIE	EEMPE	EEPE	EERE	
0x1E (0x3E)	GPIOR0	General Purpose I/O Register 0								
0x1D (0x3D)	EIMSK	-	INT6	-	-	INT3	INT2	INT1	INT0	
0x1C (0x3C)	EIFR	-	INTF6	-	-	INTF3	INTF2	INTF1	INTF0	

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Page
0x1B (0x3B)	PCIFR	-	-	-	-	-	-	-	PCIF0	
0x1A (0x3A)	Reserved	-	-	-	-	-	-	-	-	
0x19 (0x39)	TIFR4	OCF4D	OCF4A	OCF4B	-	-	TOV4	-	-	
0x18 (0x38)	TIFR3	-	-	ICF3	-	OCF3C	OCF3B	OCF3A	TOV3	
0x17 (0x37)	TIFR2	-	-	-	-	-	OCF2B	OCF2A	TOV2	
0x16 (0x36)	TIFR1	-	-	ICF1	-	OCF1C	OCF1B	OCF1A	TOV1	
0x15 (0x35)	TIFR0	-	-	-	-	-	OCF0B	OCF0A	TOV0	
0x14 (0x34)	Reserved	-	-	-	-	-	-	-	-	
0x13 (0x33)	Reserved	-	-	-	-	-	-	-	-	
0x12 (0x32)	Reserved	-	-	-	-	-	-	-	-	
0x11 (0x31)	PORTF	PORTF7	PORTF6	PORTF5	PORTF4	-	-	PORTF1	PORTF0	
0x10 (0x30)	DDRF	DDF7	DDF6	DDF5	DDF4	-	-	DDF1	DDF0	
0x0F (0x2F)	PINF	PINF7	PINF6	PINF5	PINF4	-	-	PINF1	PINF0	
0x0E (0x2E)	PORTE	-	PORTE6	-	-	-	PORTE2	-	-	
0x0D (0x2D)	DDRE	-	DDE6	-	-	-	DDE2	-	-	
0x0C (0x2C)	PINE	-	PINE6	-	-	-	PINE2	-	-	
0x0B (0x2B)	PORTD	PORTD7	PORTD6	PORTD5	PORTD4	PORTD3	PORTD2	PORTD1	PORTD0	
0x0A (0x2A)	DDRD	DDD7	DDD6	DDD5	DDD4	DDD3	DDD2	DDD1	DDD0	
0x09 (0x29)	PIND	PIND7	PIND6	PIND5	PIND4	PIND3	PIND2	PIND1	PIND0	
0x08 (0x28)	PORTC	PORTC7	PORTC6	-	-	-	-	-	-	
0x07 (0x27)	DDRC	DDC7	DDC6	-	-	-	-	-	-	
0x06 (0x26)	PINC	PINC7	PINC6	-	-	-	-	-	-	
0x05 (0x25)	PORTB	PORTB7	PORTB6	PORTB5	PORTB4	PORTB3	PORTB2	PORTB1	PORTB0	
0x04 (0x24)	DDRB	DDB7	DDB6	DDB5	DDB4	DDB3	DDB2	DDB1	DDB0	
0x03 (0x23)	PINB	PINB7	PINB6	PINB5	PINB4	PINB3	PINB2	PINB1	PINB0	
0x02 (0x22)	Reserved	-	-	-	-	-	-	-	-	
0x01 (0x21)	Reserved	-	-	-	-	-	-	-	-	
0x00 (0x20)	Reserved	-	-	-	-	-	-	-	-	

- Note:
1. For compatibility with future devices, reserved bits should be written to zero if accessed. Reserved I/O memory addresses should never be written.
 2. I/O registers within the address range \$00 - \$1F are directly bit-accessible using the SBI and CBI instructions. In these registers, the value of single bits can be checked by using the SBIS and SBIC instructions.
 3. Some of the status flags are cleared by writing a logical one to them. Note that the CBI and SBI instructions will operate on all bits in the I/O register, writing a one back into any flag read as set, thus clearing the flag. The CBI and SBI instructions work with registers 0x00 to 0x1F only.
 4. When using the I/O specific commands IN and OUT, the I/O addresses \$00 - \$3F must be used. When addressing I/O registers as data space using LD and ST instructions, \$20 must be added to these addresses. The ATmega16U4/ATmega32U4 is a complex microcontroller with more peripheral units than can be supported within the 64 location reserved in Opcode for the IN and OUT instructions. For the Extended I/O space from \$60 - \$1FF in SRAM, only the ST/STS/STD and LD/LDS/LDD instructions can be used.

5. Instruction Set Summary

Mnemonics	Operands	Description	Operation	Flags	#Clocks
ARITHMETIC AND LOGIC INSTRUCTIONS					
ADD	Rd, Rr	Add two Registers	$Rd \leftarrow Rd + Rr$	Z,C,N,V,H	1
ADC	Rd, Rr	Add with Carry two Registers	$Rd \leftarrow Rd + Rr + C$	Z,C,N,V,H	1
ADIW	RdI, K	Add Immediate to Word	$Rdh:Rdl \leftarrow Rdh:Rdl + K$	Z,C,N,V,S	2
SUB	Rd, Rr	Subtract two Registers	$Rd \leftarrow Rd - Rr$	Z,C,N,V,H	1
SUBI	Rd, K	Subtract Constant from Register	$Rd \leftarrow Rd - K$	Z,C,N,V,H	1
SBC	Rd, Rr	Subtract with Carry two Registers	$Rd \leftarrow Rd - Rr - C$	Z,C,N,V,H	1
SBCI	Rd, K	Subtract with Carry Constant from Reg.	$Rd \leftarrow Rd - K - C$	Z,C,N,V,H	1
SBIW	RdI, K	Subtract Immediate from Word	$Rdh:Rdl \leftarrow Rdh:Rdl - K$	Z,C,N,V,S	2
AND	Rd, Rr	Logical AND Registers	$Rd \leftarrow Rd \cdot Rr$	Z,N,V	1
ANDI	Rd, K	Logical AND Register and Constant	$Rd \leftarrow Rd \cdot K$	Z,N,V	1
OR	Rd, Rr	Logical OR Registers	$Rd \leftarrow Rd \vee Rr$	Z,N,V	1
ORI	Rd, K	Logical OR Register and Constant	$Rd \leftarrow Rd \vee K$	Z,N,V	1
EOR	Rd, Rr	Exclusive OR Registers	$Rd \leftarrow Rd \oplus Rr$	Z,N,V	1
COM	Rd	One's Complement	$Rd \leftarrow 0xFF - Rd$	Z,C,N,V	1
NEG	Rd	Two's Complement	$Rd \leftarrow 0x00 - Rd$	Z,C,N,V,H	1
SBR	Rd, K	Set Bit(s) in Register	$Rd \leftarrow Rd \vee K$	Z,N,V	1
CBR	Rd, K	Clear Bit(s) in Register	$Rd \leftarrow Rd \cdot (0xFF - K)$	Z,N,V	1
INC	Rd	Increment	$Rd \leftarrow Rd + 1$	Z,N,V	1
DEC	Rd	Decrement	$Rd \leftarrow Rd - 1$	Z,N,V	1
TST	Rd	Test for Zero or Minus	$Rd \leftarrow Rd \cdot Rd$	Z,N,V	1
CLR	Rd	Clear Register	$Rd \leftarrow Rd \oplus Rd$	Z,N,V	1
SER	Rd	Set Register	$Rd \leftarrow 0xFF$	None	1
MUL	Rd, Rr	Multiply Unsigned	$R1:R0 \leftarrow Rd \times Rr$	Z,C	2
MULS	Rd, Rr	Multiply Signed	$R1:R0 \leftarrow Rd \times Rr$	Z,C	2
MULSU	Rd, Rr	Multiply Signed with Unsigned	$R1:R0 \leftarrow Rd \times Rr$	Z,C	2
FMUL	Rd, Rr	Fractional Multiply Unsigned	$R1:R0 \leftarrow (Rd \times Rr) \lll 1$	Z,C	2
FMULS	Rd, Rr	Fractional Multiply Signed	$R1:R0 \leftarrow (Rd \times Rr) \lll 1$	Z,C	2
FMULSU	Rd, Rr	Fractional Multiply Signed with Unsigned	$R1:R0 \leftarrow (Rd \times Rr) \lll 1$	Z,C	2
BRANCH INSTRUCTIONS					
RJMP	k	Relative Jump	$PC \leftarrow PC + k + 1$	None	2
IJMP		Indirect Jump to (Z)	$PC \leftarrow Z$	None	2
EIJMP		Extended Indirect Jump to (Z)	$PC \leftarrow (EIND:Z)$	None	2
JMP	k	Direct Jump	$PC \leftarrow k$	None	3
RCALL	k	Relative Subroutine Call	$PC \leftarrow PC + k + 1$	None	4
ICALL		Indirect Call to (Z)	$PC \leftarrow Z$	None	4
EICALL		Extended Indirect Call to (Z)	$PC \leftarrow (EIND:Z)$	None	4
CALL	k	Direct Subroutine Call	$PC \leftarrow k$	None	5
RET		Subroutine Return	$PC \leftarrow STACK$	None	5
RETI		Interrupt Return	$PC \leftarrow STACK$	I	5
CPSE	Rd, Rr	Compare, Skip if Equal	if (Rd = Rr) $PC \leftarrow PC + 2$ or 3	None	1/2/3
CP	Rd, Rr	Compare	$Rd - Rr$	Z, N, V, C, H	1
CPC	Rd, Rr	Compare with Carry	$Rd - Rr - C$	Z, N, V, C, H	1
CPI	Rd, K	Compare Register with Immediate	$Rd - K$	Z, N, V, C, H	1
SBRC	Rr, b	Skip if Bit in Register Cleared	if (Rr(b)=0) $PC \leftarrow PC + 2$ or 3	None	1/2/3
SBRS	Rr, b	Skip if Bit in Register is Set	if (Rr(b)=1) $PC \leftarrow PC + 2$ or 3	None	1/2/3
SBIC	P, b	Skip if Bit in I/O Register Cleared	if (P(b)=0) $PC \leftarrow PC + 2$ or 3	None	1/2/3
SBIS	P, b	Skip if Bit in I/O Register is Set	if (P(b)=1) $PC \leftarrow PC + 2$ or 3	None	1/2/3
BRBS	s, k	Branch if Status Flag Set	if (SREG(s) = 1) then $PC \leftarrow PC + k + 1$	None	1/2
BRBC	s, k	Branch if Status Flag Cleared	if (SREG(s) = 0) then $PC \leftarrow PC + k + 1$	None	1/2
BREQ	k	Branch if Equal	if (Z = 1) then $PC \leftarrow PC + k + 1$	None	1/2
BRNE	k	Branch if Not Equal	if (Z = 0) then $PC \leftarrow PC + k + 1$	None	1/2
BRCS	k	Branch if Carry Set	if (C = 1) then $PC \leftarrow PC + k + 1$	None	1/2
BRCC	k	Branch if Carry Cleared	if (C = 0) then $PC \leftarrow PC + k + 1$	None	1/2
BRSH	k	Branch if Same or Higher	if (C = 0) then $PC \leftarrow PC + k + 1$	None	1/2
BRLO	k	Branch if Lower	if (C = 1) then $PC \leftarrow PC + k + 1$	None	1/2
BRMI	k	Branch if Minus	if (N = 1) then $PC \leftarrow PC + k + 1$	None	1/2
BRPL	k	Branch if Plus	if (N = 0) then $PC \leftarrow PC + k + 1$	None	1/2
BRGE	k	Branch if Greater or Equal, Signed	if (N \oplus V = 0) then $PC \leftarrow PC + k + 1$	None	1/2
BRLT	k	Branch if Less Than Zero, Signed	if (N \oplus V = 1) then $PC \leftarrow PC + k + 1$	None	1/2
BRHS	k	Branch if Half Carry Flag Set	if (H = 1) then $PC \leftarrow PC + k + 1$	None	1/2
BRHC	k	Branch if Half Carry Flag Cleared	if (H = 0) then $PC \leftarrow PC + k + 1$	None	1/2
BRTS	k	Branch if T Flag Set	if (T = 1) then $PC \leftarrow PC + k + 1$	None	1/2
BRTC	k	Branch if T Flag Cleared	if (T = 0) then $PC \leftarrow PC + k + 1$	None	1/2
BRVS	k	Branch if Overflow Flag is Set	if (V = 1) then $PC \leftarrow PC + k + 1$	None	1/2

Mnemonics	Operands	Description	Operation	Flags	#Clocks
BRVC	k	Branch if Overflow Flag is Cleared	if (V = 0) then PC ← PC + k + 1	None	1/2
BRIE	k	Branch if Interrupt Enabled	if (I = 1) then PC ← PC + k + 1	None	1/2
BRID	k	Branch if Interrupt Disabled	if (I = 0) then PC ← PC + k + 1	None	1/2
BIT AND BIT-TEST INSTRUCTIONS					
SBI	P,b	Set Bit in I/O Register	I/O(P,b) ← 1	None	2
CBI	P,b	Clear Bit in I/O Register	I/O(P,b) ← 0	None	2
LSL	Rd	Logical Shift Left	Rd(n+1) ← Rd(n), Rd(0) ← 0	Z,C,N,V	1
LSR	Rd	Logical Shift Right	Rd(n) ← Rd(n+1), Rd(7) ← 0	Z,C,N,V	1
ROL	Rd	Rotate Left Through Carry	Rd(0) ← C, Rd(n+1) ← Rd(n), C ← Rd(7)	Z,C,N,V	1
ROR	Rd	Rotate Right Through Carry	Rd(7) ← C, Rd(n) ← Rd(n+1), C ← Rd(0)	Z,C,N,V	1
ASR	Rd	Arithmetic Shift Right	Rd(n) ← Rd(n+1), n=0..6	Z,C,N,V	1
SWAP	Rd	Swap Nibbles	Rd(3..0) ← Rd(7..4), Rd(7..4) ← Rd(3..0)	None	1
BSET	s	Flag Set	SREG(s) ← 1	SREG(s)	1
BCLR	s	Flag Clear	SREG(s) ← 0	SREG(s)	1
BST	Rr, b	Bit Store from Register to T	T ← Rr(b)	T	1
BLD	Rd, b	Bit load from T to Register	Rd(b) ← T	None	1
SEC		Set Carry	C ← 1	C	1
CLC		Clear Carry	C ← 0	C	1
SEN		Set Negative Flag	N ← 1	N	1
CLN		Clear Negative Flag	N ← 0	N	1
SEZ		Set Zero Flag	Z ← 1	Z	1
CLZ		Clear Zero Flag	Z ← 0	Z	1
SEI		Global Interrupt Enable	I ← 1	I	1
CLI		Global Interrupt Disable	I ← 0	I	1
SES		Set Signed Test Flag	S ← 1	S	1
CLS		Clear Signed Test Flag	S ← 0	S	1
SEV		Set Twos Complement Overflow.	V ← 1	V	1
CLV		Clear Twos Complement Overflow	V ← 0	V	1
SET		Set T in SREG	T ← 1	T	1
CLT		Clear T in SREG	T ← 0	T	1
SEH		Set Half Carry Flag in SREG	H ← 1	H	1
CLH		Clear Half Carry Flag in SREG	H ← 0	H	1
DATA TRANSFER INSTRUCTIONS					
MOV	Rd, Rr	Move Between Registers	Rd ← Rr	None	1
MOVW	Rd, Rr	Copy Register Word	Rd+1:Rd ← Rr+1:Rr	None	1
LDI	Rd, K	Load Immediate	Rd ← K	None	1
LD	Rd, X	Load Indirect	Rd ← (X)	None	2
LD	Rd, X+	Load Indirect and Post-Inc.	Rd ← (X), X ← X + 1	None	2
LD	Rd, -X	Load Indirect and Pre-Dec.	X ← X - 1, Rd ← (X)	None	2
LD	Rd, Y	Load Indirect	Rd ← (Y)	None	2
LD	Rd, Y+	Load Indirect and Post-Inc.	Rd ← (Y), Y ← Y + 1	None	2
LD	Rd, -Y	Load Indirect and Pre-Dec.	Y ← Y - 1, Rd ← (Y)	None	2
LDD	Rd, Y+q	Load Indirect with Displacement	Rd ← (Y + q)	None	2
LD	Rd, Z	Load Indirect	Rd ← (Z)	None	2
LD	Rd, Z+	Load Indirect and Post-Inc.	Rd ← (Z), Z ← Z+1	None	2
LD	Rd, -Z	Load Indirect and Pre-Dec.	Z ← Z - 1, Rd ← (Z)	None	2
LDD	Rd, Z+q	Load Indirect with Displacement	Rd ← (Z + q)	None	2
LDS	Rd, k	Load Direct from SRAM	Rd ← (k)	None	2
ST	X, Rr	Store Indirect	(X) ← Rr	None	2
ST	X+, Rr	Store Indirect and Post-Inc.	(X) ← Rr, X ← X + 1	None	2
ST	-X, Rr	Store Indirect and Pre-Dec.	X ← X - 1, (X) ← Rr	None	2
ST	Y, Rr	Store Indirect	(Y) ← Rr	None	2
ST	Y+, Rr	Store Indirect and Post-Inc.	(Y) ← Rr, Y ← Y + 1	None	2
ST	-Y, Rr	Store Indirect and Pre-Dec.	Y ← Y - 1, (Y) ← Rr	None	2
STD	Y+q, Rr	Store Indirect with Displacement	(Y + q) ← Rr	None	2
ST	Z, Rr	Store Indirect	(Z) ← Rr	None	2
ST	Z+, Rr	Store Indirect and Post-Inc.	(Z) ← Rr, Z ← Z + 1	None	2
ST	-Z, Rr	Store Indirect and Pre-Dec.	Z ← Z - 1, (Z) ← Rr	None	2
STD	Z+q, Rr	Store Indirect with Displacement	(Z + q) ← Rr	None	2
STS	k, Rr	Store Direct to SRAM	(k) ← Rr	None	2
LPM		Load Program Memory	R0 ← (Z)	None	3
LPM	Rd, Z	Load Program Memory	Rd ← (Z)	None	3
LPM	Rd, Z+	Load Program Memory and Post-Inc	Rd ← (Z), Z ← Z+1	None	3
ELPM		Extended Load Program Memory	R0 ← (RAMPZ:Z)	None	3
ELPM	Rd, Z	Extended Load Program Memory	Rd ← (Z)	None	3
ELPM	Rd, Z+	Extended Load Program Memory	Rd ← (RAMPZ:Z), RAMPZ:Z ← RAMPZ:Z+1	None	3

Mnemonics	Operands	Description	Operation	Flags	#Clocks
SPM		Store Program Memory	(Z) ← R1:R0	None	-
IN	Rd, P	In Port	Rd ← P	None	1
OUT	P, Rr	Out Port	P ← Rr	None	1
PUSH	Rr	Push Register on Stack	STACK ← Rr	None	2
POP	Rd	Pop Register from Stack	Rd ← STACK	None	2
MCU CONTROL INSTRUCTIONS					
NOP		No Operation		None	1
SLEEP		Sleep	(see specific description for Sleep function)	None	1
WDR		Watchdog Reset	(see specific description for WDR/timer)	None	1
BREAK		Break	For On-chip Debug Only	None	N/A

6. Ordering Information

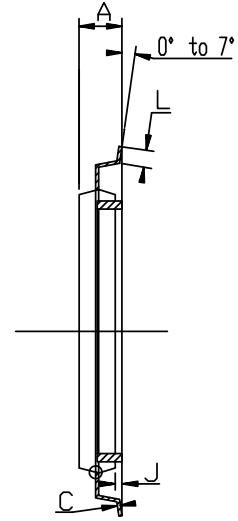
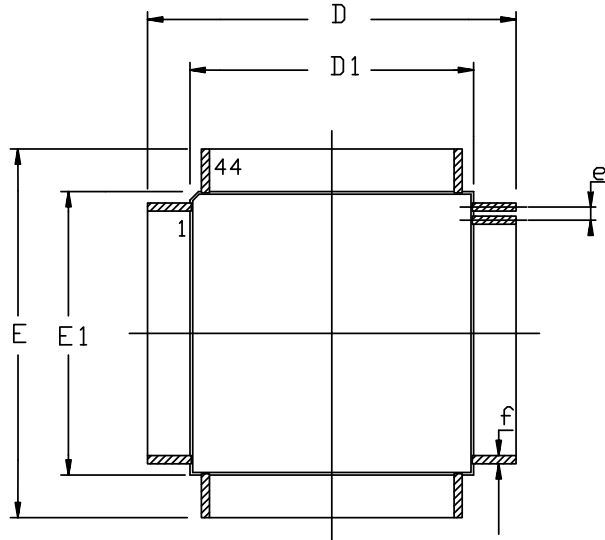
Table 6-1. Possible Order Entries

Ordering Code	USB interface	Speed (MHz)	Power Supply (V)	Package	Operation Range	Product Marking
ATmega32U4-16AU	Device only	8-16	2.7 - 5.5	TQFP44	Industrial (-40° to +85°C) Green	mega32U4-16AU
ATmega32U4-16MU	Device only	8-16	2.7 - 5.5	QFN44	Industrial (-40° to +85°C) Green	mega32U4-16MU

7. Package Information

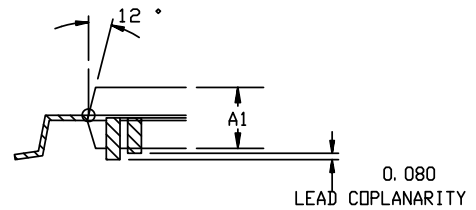
Package Type	
ML	ML, 44 - Lead, 10 x 10 mm Body Size, 1.0 mm Body Thickness 0.8 mm Lead Pitch, Thin Profile Plastic Quad Flat Package (TQFP)
PW	PW, 44 - Lead 7.0 x 7.0 mm Body, 0.50 mm Pitch Quad Flat No Lead Package (QFN)

7.1 TQFP44



COMMON DIMENSIONS IN MM


SYMBOL	Min	Max	NOTES
A	----	1.20	
A1	0.95	1.05	
C	0.09	0.20	
D	12.00 BSC		
D1	10.00 BSC		
E	12.00 BSC		
E1	10.00 BSC		
J	0.05	0.15	
L	0.45	0.75	
e	0.80 BSC		
f	0.30	0.45	



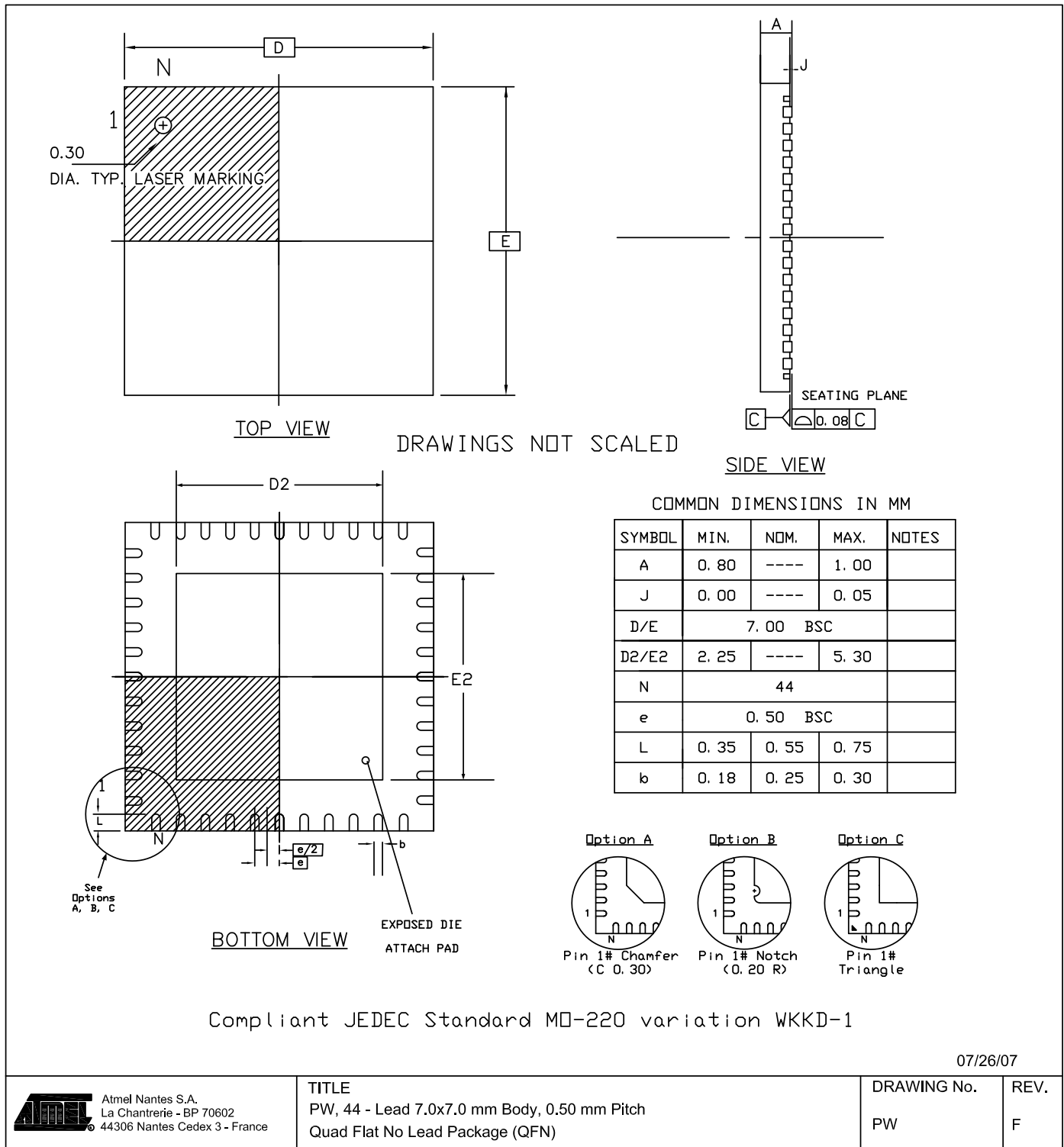
07/27/07

NOTES: STANDARD NOTES FOR PQFP/VQFP/TQFP/DQFP

1. DIMENSIONING & TOLERANCING CONFORM TO ASME Y14.5M. – 1982.
2. "D1 AND E1" DIMENSIONS DO NOT INCLUDE MOLD PROTUSIONS
MOLD PROTUSIONS SHALL NOT EXCEED 0.25 mm (0.010 INCH) .
THE TOP PACKAGE BODY SIZE MAY BE SMALLER THAN THE BOTTOM
PACKAGE BODY SIZE BY AS MUCH AS 0.15 mm.
3. DATUM PLANE "H" LOCATED AT MOLD PARTING LINE AND
COINCIDENT WITH LEAD, WHERE LEAD EXISTS PLASTIC BODY AT
BOTTOM OF PARTING LINE.
4. DATUM "A" AND "D" TO BE DETERMINED AT DATUM PLANE H.
5. DIMENSION "f" DOES NOT INCLUDE DAMBAR PROTUSION ALLOWABLE
DAMBAR PROTUSION SHALL BE 0.08 mm/.003" TOTAL EXCESS OF THE
"f" DIMENSION AT MAXIMUM MATERIAL CONDITION.
DAMBAR CANNOT BE LOCATED ON THE LOWER RADIUS OR THE FOOT.

 <p>Atmel Nantes S.A. La Chantrerie - BP 70602 44306 Nantes Cedex 3 - France</p>	TITLE	DRAWING No.	REV.
---	-------	-------------	------

7.2 QFN44



8. Errata

The revision letter in this section refers to the revision of the ATmega16U4/ATmega32U4 device.

8.1 ATmega16U4/ATmega32U4 Rev A

1. Spike on TWI pins when TWI is enabled

100 ns negative spike occurs on SDA and SCL pins when TWI is enabled.

Problem Fix/work around

No known work around, enable ATmega16U4/ATmega32U4 TWI first versus the others nodes of the TWI network.

2. High current consumption in sleep mode

If a pending interrupt cannot wake the part up from the selected mode, the current consumption will increase during sleep when executing the SLEEP instruction directly after a SEI instruction.

Problem Fix/work around

Before entering sleep, interrupts not used to wake up the part from the sleep mode should be disabled.

3. Extra power consumption

The typical power consumption is increased by about 30 μ A in power-down mode.

Problem Fix/work around

None.

4. Internal RC oscillator start up issue.

When the part is configured to start on internal RC, the oscillator may not start properly after power-on.

Problem Fix/work around

Do not configure the part to start with the internal oscillator (default part configuration is to start with the external crystal oscillator).

5. Internal RC oscillator calibration issue.

The default internal RC oscillator frequency may be lower than 8MHz.

Problem Fix/work around

Parts are configured so that the internal RC oscillator frequency is as close as possible to the 8MHz default target frequency.

9. Datasheet Revision History for ATmega16U4/ATmega32U4

Please note that the referring page numbers in this section are referred to this document. The referring revision in this section are referring to the document revision.

9.1 Revision A.

1. Initial document version.

9.2 Revision B.

1. Added ATmega16U4 device.
2. Created errata section and added ATmega16U4.
3. Update High Speed Timer, asynchronous description [Section 15. on page 139](#).



Headquarters

Atmel Corporation
2325 Orchard Parkway
San Jose, CA 95131
USA
Tel: 1(408) 441-0311
Fax: 1(408) 487-2600

International

Atmel Asia
Room 1219
Chinachem Golden Plaza
77 Mody Road Tsimshatsui
East Kowloon
Hong Kong
Tel: (852) 2721-9778
Fax: (852) 2722-1369

Atmel Europe
Le Krebs
8, Rue Jean-Pierre Timbaud
BP 309
78054 Saint-Quentin-en-
Yvelines Cedex
France
Tel: (33) 1-30-60-70-00
Fax: (33) 1-30-60-71-11

Atmel Japan
9F, Tonetsu Shinkawa Bldg.
1-24-8 Shinkawa
Chuo-ku, Tokyo 104-0033
Japan
Tel: (81) 3-3523-3551
Fax: (81) 3-3523-7581

Product Contact

Web Site
www.atmel.com

Technical Support
avr@atmel.com

Sales Contact
www.atmel.com/contacts

Literature Requests
www.atmel.com/literature

Disclaimer: The information in this document is provided in connection with Atmel products. No license, express or implied, by estoppel or otherwise, to any intellectual property right is granted by this document or in connection with the sale of Atmel products. **EXCEPT AS SET FORTH IN ATMEL'S TERMS AND CONDITIONS OF SALE LOCATED ON ATMEL'S WEB SITE, ATMEL ASSUMES NO LIABILITY WHATSOEVER AND DISCLAIMS ANY EXPRESS, IMPLIED OR STATUTORY WARRANTY RELATING TO ITS PRODUCTS INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. IN NO EVENT SHALL ATMEL BE LIABLE FOR ANY DIRECT, INDIRECT, CONSEQUENTIAL, PUNITIVE, SPECIAL OR INCIDENTAL DAMAGES (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF PROFITS, BUSINESS INTERRUPTION, OR LOSS OF INFORMATION) ARISING OUT OF THE USE OR INABILITY TO USE THIS DOCUMENT, EVEN IF ATMEL HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.** Atmel makes no representations or warranties with respect to the accuracy or completeness of the contents of this document and reserves the right to make changes to specifications and product descriptions at any time without notice. Atmel does not make any commitment to update the information contained herein. Unless specifically provided otherwise, Atmel products are not suitable for, and shall not be used in, automotive applications. Atmel's products are not intended, authorized, or warranted for use as components in applications intended to support or sustain life.

© 2008 Atmel Corporation. All rights reserved. Atmel®, logo and combinations thereof, and others are registered trademarks or trademarks of Atmel Corporation or its subsidiaries. Other terms and product names may be trademarks of others.



Plagiarism Checker X Originality Report

Similarity Found: 0%

Date: miércoles, septiembre 05, 2018

Statistics: 0 words Plagiarized / 7080 Total words

Remarks: No Plagiarism Detected - Your Document is Healthy.

RESUMEN En este proyecto se diseñó e implementó un prototipo electrónico de bajo costo para lograr la interacción con proyectores. Básicamente un sistema interactivo es un sistema de seguimiento de una fuente de luz (normalmente un lápiz infrarrojo) el cual es relacionado con la posición del cursor del ratón del sistema operativo del computador desde el cual se envía la información, es decir, se logra la interacción con una imagen proyectada "simulando" la posición del ratón del sistema operativo en base a la posición de un lápiz infrarrojo en la pantalla.

Generalmente la interacción con proyectores es un accesorio costoso y específico para la marca del equipo. Con la facilidad de desarrollo de prototipos en plataformas abiertas (PICs, Arduino, Raspberry Pi, etc.) la construcción de un dispositivo que "ubique" una fuente de luz infrarroja y la relacione con la posición relativa al respecto de la pantalla es sencillo de realizar empleando componentes de fácil adquisición y de forma local.

En este proyecto se presenta el diseño e implementación de un prototipo de un sistema de seguimiento de luz infrarroja capaz de enviar la información de ubicación relativa a la pantalla. Esta información es interpretada por el sistema operativo de un computador como la posición del ratón, logrando así la interactividad con contenido proyectado.

Palabras clave: Posicionamiento Infrarrojo Ratón Interacción

ABSTRACT In this project a low-cost prototype for whiteboard type interaction with a projected screen was designed and the built. Basically, an interactive projector uses a light source (typically an infrared pen) as an input to calculate its relative position to the projected image.

It uses the relative position of the infrared pen to simulate the mouse's cursor position by Operating System and so controlling how the mouse moves (and clicks). Usually, this is commercially done with expensive equipment which typically is vendor locked. With open platforms such as Arduino, Microchip PIC, Raspberry Pi, etc. building a device capable of "finding" the relative coordinates (x, y) of a light source, process and send them to the computer can be achieved easily with low cost components. In this project the designing and building of the device is discussed, along with the theoretical basis and algorithm development.

Keywords: Interactive Infrared Positioning Mouse

INTRODUCCIÓN Actualmente, el hombre está acostumbrado a interactuar con sus dispositivos electrónicos en prácticamente todos los aspectos de la vida. El uso de los teléfonos inteligentes con pantalla táctil y el uso de tecnologías "touch" en otros dispositivos obligan a buscar un uso sencillo e intuitivo en casi todas las actividades cotidianas.

Una de las actividades, que todavía no ha "migrado" extensamente a esta interacción, es la proyección de diapositivas o de información de texto. Para interactuar en dichas ocasiones hay varias soluciones en el mercado cada una con sus pro y contras, por ejemplo: pantallas táctiles – Microsoft Surface Hub, proyectores interactivos – NEC, Epson, Optoma, dispositivos de interfaz – Tablet Wacom, entre otras.

Entre las ventajas de usar un proyector diseñado para la interacción es que se cuenta con soporte de fábrica, stock de repuestos, actualizaciones de software. Las desventajas de dichos sistemas propietarios suelen ser su alto costo y que no existe compatibilidad entre marcas, es decir, si se adquiere un proyector marca A los accesorios de marca B no son compatibles. Los monitores de pantalla táctil son de los más costosos y limitados en tamaño.

Pues usan dispositivos infrarrojos para detección o son construidos con técnicas similares a las usadas en teléfonos móviles (pantalla táctil resistiva o capacitiva). Las Tablet digitalizadoras (tipo Wacom) son dispositivos diseñados para "dibujar" a mano en el PC, permitiendo la interacción de diseñadores gráficos con programas especializados, además de que requieren controladores específicos.

Otra solución son los proyectores interactivos, los cuales usan tecnología propietaria de cada fabricante que puede ser infrarroja, bluetooth, radiofrecuencia, etc. que son únicamente compatibles con equipos del mismo fabricante. El sistema operativo de una computadora es capaz de procesar la información de coordenadas enviadas por un ratón (mouse) y desplegar la posición relativa del cursor basado en dichas coordenadas.

Usando un módulo de posicionamiento infrarroja capaz de ubicar, en coordenadas relativas de posición, una fuente de luz infrarroja, procesando dichas coordenadas de manera que el sistema operativo sea capaz de interpretarlas a manera de movimiento de ratón.

ANTECEDENTES En múltiples ambientes (comerciales, educativos y similares) donde se requiere presentar información a una audiencia de varias personas normalmente se proyecta el contenido, el cual puede ser multimedia, diapositivas, documentos, etc.

En más de una ocasión es necesario interactuar con dicho contenido: hacer anotaciones, resaltar alguna idea o simplemente "dibujar" algo en la proyección para aclarar la idea presentada. Generalmente cuando esto es requerido, dependiendo del tipo de superficie o pantalla donde esté siendo proyectada la imagen, se usan marcadores de tinta, lápices o dispositivos interactivos que no rayan la superficie.

En general, la mayoría de los proyectores son tradicionales; es decir, no soportan dispositivos interactivos, lo cual implica que no es posible interactuar de manera eficiente con la información proyectada. A veces, el presentador prefiere proyectar la imagen sobre una pizarra para poder interactuar con lo presentado. Pero esto no es óptimo pues la pizarra tiene un brillo propio que distorsiona la presentación a una parte de la audiencia o simplemente el marcador empleado es de un color que no es posible observar por ciertos miembros de la audiencia, dañando el objetivo de interactuar del presentador. En la figura 0.1.

abajo se puede observar que se debe usar la pantalla de proyección (Figura 1a) para evitar el brillo o la proyección con brillo (Figura 1b) – cuando no se usa la pantalla – que impide que todo el contenido se pueda observar correctamente. // Figura. 1. (1a) Proyección con pantalla (1b) Proyección sin pantalla (con brillo) Fuente: Elaborado por el autor Usar un proyector interactivo es una gran solución para cuando es requerida la interacción con la información proyectada, con el limitante de que generalmente es costoso y requiere equipo especializado, algunas veces incluso con software propio del fabricante. Lo que limita su uso a ambientes muy específicos.

En varios centros educativos del país existen proyectores en las aulas, pero dichos proyectores, al ser estándar no permiten interacción alguna. Reemplazar dichos equipos por proyectores interactivos tendría un costo muy alto, pues se debe prorratear el costo del proyector, el software, los insumos, instalación, calibración, etc. Eso puede dejar fuera del presupuesto de la mayoría de las instituciones usar proyectores interactivos.

Un sencillo dispositivo electrónico capaz de ubicar una fuente de luz infrarroja es una solución a los inconvenientes de interacción en proyectores estándar, pues permite de una manera sencilla, económica y no especializada (no requiere la instalación de software) de interactuar con la información presentada, facilitando la presentación.

Este proyecto incluye el diseño y construcción de un prototipo de dispositivo capaz de

realizar la ubicación relativa de una fuente de luz al respecto de la información proyectada, procesar dicha ubicación y enviarla como un dispositivo de interfaz humana (Human Interface Device) a un computador – independiente del sistema operativo – para lograr interacción con cualquier tipo de proyector.

El dispositivo usará programación con software abierto, por lo que será posible realizar cualquier tipo de programa orientado a mejorar la experiencia de aprendizaje y explicación en clases o cualquier otro uso que se le quiera dar.

ALCANCE En el presente proyecto se plantea las bases teóricas del funcionamiento del prototipo del dispositivo, el algoritmo para lograr una funcionalidad básica basada en simular el movimiento del cursor del ratón de un computador usando un lápiz infrarrojo y una manera sencilla de implementarlo con componentes de fácil adquisición en el mercado nacional.

Funciones avanzadas, tales como: aplicaciones de software para PC, calibración, proyección o diseño de monturas a la medida para la instalación no serán realizadas. Al ser un prototipo el dispositivo será sencillo de implementar y podrá ser usado como base para el desarrollo de un dispositivo comercial más elaborado, con manufactura profesional e interfaz de calibración para computador.

OBJETIVOS Objetivo general: Crear un dispositivo para transformar en interactivo un proyector normal.

Objetivos específicos: Definir los elementos constitutivos de un dispositivo electrónico para transformar un proyector estándar en interactivo. Elaborar el diagrama de bloques para el funcionamiento del dispositivo. Diseñar el circuito electrónico para la implementación del dispositivo. Desarrollar el algoritmo de programación para el funcionamiento del dispositivo. Implementar el dispositivo.

Realizar la validación del dispositivo a través de pruebas de uso.

DESCRIPCIÓN DEL CONTENIDO DE LOS CAPÍTULOS Capítulo 1: en este capítulo se presenta el marco teórico relacionado con el dispositivo desde el punto de vista para su elaboración.

No se incluirán detalles de qué es la luz infrarroja, cómo funciona un ratón, qué es un USB y similares, pues se asume que el lector está familiarizado con las tecnologías empleadas y más bien, se hará una breve explicación de los principios sobre los cuales funciona el dispositivo. Capítulo 2: en este capítulo se evalúan las opciones de diseño e implementación del prototipo, se analiza brevemente los componentes y las opciones para su construcción.

Capítulo 3: en este capítulo se documenta el proceso de construcción empleado, así como el algoritmo de programación desarrollado para lograr la operación de este. Así mismo se resume las pruebas realizadas de operación, y se documentan las experiencias de implementación. Conclusiones y recomendaciones: como su nombre lo indica se presentan los resultados finales del proyecto, los objetivos alcanzados y algunas recomendaciones para mejora y trabajos futuros.

CAPÍTULO 1 FUNDAMENTACIÓN TEÓRICA 1.1: MARCO TEÓRICO Y CONCEPTUAL En este capítulo se revisan las tecnologías que permitieron implementar el dispositivo para lograr la interactividad con un proyector, con esta información luego se continúa con el diseño e implementación de este.

Dado que el sistema está basado en microcontrolador primero se analiza el tipo de microcontrolador usado, para luego revisar dispositivos de entrada de información y de salida. 1.2.: MICROCONTROLADOR Un microcontrolador es básicamente un computador completo, pues incluye: Entradas Unidad de procesamiento o CPU Memoria RAM y ROM Salidas Los microcontroladores son circuitos integrados capaces de ser programados, incluyen todo lo requerido para ejecutar su programa en una sola placa electrónica.

Existen varios tipos de dispositivos microcontroladores diseñados para múltiples funciones y también de uso general. Dada la gran variedad de dispositivos en el mercado, y teniendo en cuenta sus características, una de las plataformas más populares para implementación de prototipos es Arduino. Este dispositivo surgió del movimiento de "open-software" y "open-hardware". Fue creado inicialmente en Italia alrededor del año 2005. En la tabla 1.1

se observa una breve comparación entre controladores. Tabla. 1. 1. Cuadro comparativo entre microcontroladores _PIC Microchip _Arduino _ _Tiene ambiente propio de programación _Si _Si _ _Se puede programar múltiples veces _Si _Si _ _Permite programación sin removerlo de la placa _No _Si _ _Tiene puerto serie embebido, no requiere de hardware extra para operación _No _Si _ _Existen periféricos compatibles entre sí _Si _Si _ _Fuente: Elaborado por el autor Arduino está basado en distintos tipos de microcontrolador (depende del modelo y características a usar), aunque generalmente se usa el ATmega.

Existen varios tipos de placas Arduino, cada una diseñada para diversos usos, puntualmente – para desarrollo de este proyecto – serán empleados dos tipos: Arduino Pro Mini Arduino Pro Micro ¿Por qué usar dos microcontroladores para un proyecto? Pues bien, se revisa primero las características de cada uno: Arduino Pro Mini: Este dispositivo está fabricado en una placa pequeña, está orientado a proyectos donde el espacio es imperativo y consta de un microprocesador ATmega328 (la hoja de especificaciones técnicas está en el Anexo 3).

Existen dos versiones: una que trabaja con 5 voltios y resonador de 16 MHz y una de 3,3 voltios y resonador de 8 MHz. Las especificaciones técnicas de este equipo se detallan en la tabla 1.2.: Tabla. 1. 2. Características del Arduino Pro Mini Microprocesador

_ATMega 328 _ Fuente de poder _3,35 a 12v (el modelo de 3,3v) y de 5 a 12 v (el modelo de 5v) _ Voltaje de operación del circuito _3,3 a 5v (depende del modelo) _ Pines de entrada/salida digital _14 _ Pines PWM _6 _ UART _1 _ SPI _1 _ I2C _1 _ Pines de entrada analógica _6 _ Interrupciones externas _2 _ Corriente máxima por pin I/O _40 mA _ Memoria Flash _32 KB, (2KB son usados por bootloader) _ SRAM _2 KB _ EEPROM _1 KB _ Velocidad del reloj _8 MHz (la versión de 3,3v) y 16 MHz (la versión de 5v) _ Fuente: Elaborado por el autor Si se observan las características del dispositivo se puede notar que no cuenta con puerto de comunicación serie a bordo, por lo que es necesario usar una tarjeta adicional para poderlo programar de manera sencilla.

Arduino Pro Micro: Este dispositivo es muy similar al Arduino Pro Mini, de igual manera está fabricado en una placa pequeña y la gran diferencia (con el Pro Mini) es que usa un microprocesador ATMega32U4 que si tiene puerto serie embebido (la hoja de especificaciones técnicas está en el Anexo 4). De igual manera que el Pro Mini, existen dos versiones: una que trabaja con 5 voltios y resonador de 16 MHz y una de 3,3 voltios y resonador de 8 MHz.

Las especificaciones técnicas de este equipo se detallan en la tabla 1.3.: Tabla. 1. 3. Características del Arduino Pro Micro Microprocesador _ATMega 32U4 _ Fuente de poder _4 a 12v (el modelo de 3,3v) y de 5 a 12 v (el modelo de 5v) _ Voltaje de operación del circuito _3,3 a 5v (depende del modelo) _ Pines de entrada/salida digital _12 _ Pines PWM _5 _ Puerto de comunicación _1 Serie Tx-Rx en hardware con puerto micro-USB _ SPI _1 _ I2C _1 _ Pines de entrada analógica _4 _ Interrupciones externas _2 _ Corriente máxima por pin I/O _40 mA _ Memoria Flash _32 KB, (2KB son usados por bootloader) _ SRAM _2,5 KB _ EEPROM _1 KB _ Velocidad del reloj _8 MHz (la versión de 3,3v) y 16 MHz (la versión de 5v) _ Fuente: Elaborado por el autor Este dispositivo cuenta con un puerto micro-USB implementado en la placa, facilitando así la programación y uso del dispositivo.

Ambos dispositivos son muy similares físicamente, tienen un diseño pequeño como para ser implementados en una placa de circuito impreso pequeña, la gran diferencia entre los dos es que el Pro Mini no tiene puerto de comunicación en la placa y el Pro Micro sí. 1.3: DISPOSITIVOS DE ENTRADA (MÓDULO DE POSICIONAMIENTO INFRARROJO) Para lograr la interactividad se usan dos dispositivos como entrada del sistema.

Ambos son dispositivos digitales, un módulo de posicionamiento y un lápiz infrarrojos. El diagrama de bloques del dispositivo se observa en la figura 2.2 en el capítulo dos. Módulo de posicionamiento infrarrojo: Este dispositivo es la parte más importante del

proyecto, pues, es el que realiza el seguimiento y envía la información de posicionamiento de la fuente de luz infrarroja para su posterior procesamiento en el Arduino Pro Micro.

El módulo básicamente es un dispositivo de posicionamiento IR capaz de hacer el seguimiento y posicionamiento de hasta cuatro fuentes de luz infrarroja de manera simultánea, enviando la información de ubicación relativa (coordenadas x, y) y el tamaño circular (blob) de la fuente infrarroja. Esto puede ser usado en caso de que se desee usar más de un lápiz infrarrojo de manera simultánea.

Pero, por limitaciones del sistema operativo, no es recomendable usar más de un ratón al tiempo. Este dispositivo contiene un módulo de semiconductor CMOS que cuenta con un filtro pasa-banda infrarrojo (de allí que solo es "sensible" a la luz infrarroja), cuenta con DSP (Digital Signal Processor) y entrega la información digital en protocolo I2C para su procesamiento. En la tabla 1.4.,

se muestran algunas especificaciones técnicas de este dispositivo (para obtenerlas todas el fabricante - PixArt - requiere una compra inicial de 1000 unidades y firmar un acuerdo de confidencialidad, por lo que únicamente están disponibles algunas): Tabla. 1. 4.

Características del módulo de seguimiento infrarrojo Sensor de imagen _CMOS _
_Voltaje de operación _2,0 a 3,6v _Consumo máximo de corriente _44 mA _Reloj del sistema _De 5 a 23 MHz _Interfaz _I2C _Resolución _128 x 96 pixeles _Número de objetos a seguir _Máximo 4 simultáneos _Capaz de entregar información de posición _Sí, mediante I2C en coordenadas X y Y _Ángulo de visión vertical _23° _Ángulo de visión horizontal _33° _Distancia de detección _0 a 3 metros _Fuente: Elaborado por el autor El protocolo mediante el cual este módulo envía los datos es I2C o Inter-Integrated Circuit (Entre circuitos integrados) el cual es un bus serie inventado por Philips Semiconductor en 1982.

El protocolo fue diseñado inicialmente para permitir el envío de datos entre periféricos de baja velocidad y microcontroladores en una corta distancia en una placa de circuito impreso. El protocolo usa dos líneas bidireccionales de datos: Línea de Datos Seriales (Serial Data Line o SDA) Línea de Reloj Serie (Serial Clock Line o SCL) Las líneas de datos generalmente funcionan con lógica de +5 ó +3,3 voltios, es de prioridad (un dispositivo es el maestro y los demás esclavos) y usa velocidades de 100 kbit/s.

El número total de nodos está limitado por la tabla de direcciones (usa direccionamiento de 7 bits) y la capacitancia total del bus que es de 400 pF. Dicha impedancia implica que se requiere usar una referencia de tierra común en toda la comunicación. Lápiz infrarrojo: Este dispositivo es la fuente de luz infrarroja. Es un

circuito serie sencillo, con un diodo LED infrarrojo, una resistencia de 220 Ω (usada como limitadora de corriente) un interruptor miniatura y una pila o batería, su diagrama esquemático se observa en la figura 2.7 del Capítulo 2.

Al cerrar el interruptor se activa el circuito encendiendo el LED infrarrojo, generando así la fuente de luz.

1.4: DISPOSITIVOS DE SALIDA – INTERCONEXIÓN CON LA COMPUTADORA

Para la interconexión con la computadora se usa el puerto USB integrado en el Arduino Pro Micro, conectándolo mediante un cable de micro-USB (terminal tipo Micro-B) a USB (Terminal tipo A).

El USB (Bus Serie Universal o Universal Serial Bus) es un estándar de la industria de computadores desarrollado para conectar, comunicar y enviar energía entre distintos dispositivos (por ejemplo: la computadora y periféricos). Es un protocolo de comunicación serial diseñado para facilitar la interconexión de periféricos a la computadora. El sistema consiste en un host con uno o más puertos de comunicación "hacia abajo" o downstream, esto forma una configuración de topología tipo estrella, un host USB puede soportar un máximo de 127 dispositivos. El estándar de USB 2.0

permite una distancia máxima entre host de 5 metros si el dispositivo está funcionando a 480 Mbit/s y funciona a 5 voltios con una tolerancia de $\pm 5\%$.

CAPÍTULO 2 PROPUESTA El prototipo, al ser un dispositivo digital, consta de entradas, procesador y salidas; con un diagrama de bloques como se observa en la figura 2.2: / Figura. 2. 2.

Diagrama de bloques del prototipo Fuente: Elaborado por el autor Así entonces, se desarrolla cada bloque para su posterior construcción, como se explica en las siguientes secciones. 2.1: DIAGRAMA ESQUEMÁTICO DEL DISPOSITIVO Como entrada para el sistema se emplea el módulo de posicionamiento IR, el cual es un dispositivo que cuenta con 8 pines, según se observa en la figura 2.3: / Figura. 2. 3.

Detalle del módulo de posicionamiento Fuente: (Hobley, 2009) Como se observa, el módulo solamente soporta alimentación de 3,3 voltios; por ende, la línea de datos I2C debe ser también de 3,3 voltios. Es por esto que se emplean varios controladores Arduino de 3,3 voltios para no hacer conversión del nivel de la señal. Para que el dispositivo funcione se deben tener dos señales: El pin de reset debe estar a nivel uno lógico (1L), lo cual se logra mediante un capacitor de 1 μ F conectado a tierra (que se descargará al apagar el equipo) y una resistencia de 30k Ω usada como limitadora de corriente.

Una señal de reloj digital de al menos 5 MHz, para lograr que el módulo funcione. Para generar la señal de reloj es posible usar un reloj digital (no únicamente el cristal, pues la señal debe ser acondicionada) o también un Arduino Pro Mini como fuente aprovechando que uno de sus pines físicos está conectado a la señal de reloj del microprocesador ATmega.

Si bien, el Arduino Pro Mini es capaz de procesar la señal I2C enviada por el módulo, éste no cuenta con puerto serie físico y no es capaz de enviar la señal de posición del ratón USB a la computadora. El Arduino Pro Micro, en cambio, no tiene un pin físico conectado a la señal de reloj del microprocesador, he ahí el por qué se emplean dos microcontroladores, el Arduino Pro Mini es el reloj y el Pro Micro es el procesador de la señal.

Según el diagrama de pines del Arduino Pro Mini en el pin digital 8 se tiene una conexión a la señal interna de reloj del ATmega, según se observa en la figura 2.4: / Figura. 2. 4. Diagrama de pines del Arduino Pro Mini Fuente: (Sparkfun Electronics, 2016) El pin D8 indica que tiene conexión CLKO, es decir la señal de "ClockOut" o "Salida de Reloj", para usar dicha salida se debe modificar el código bootloader del Arduino; es decir, se modifica la programación básica de la EEPROM del microcontrolador para lograr que el dispositivo entregue la señal de reloj. Este proceso se indica en el Capítulo 3 en la implementación del dispositivo.

La ventaja del bus I2C es su sencillez de conexión, pues se requieren dos pines de conexión: SDA (Serial Data o Dato Serie) y SCL (System Clock o Reloj del Sistema), los cuales se conectan a los pines correspondientes en el Arduino Pro Micro. Dichos pines corresponden a los pines digitales 2 y 3; como se observa en la figura 2.5. / Figura 2. 5.

Diagrama de pines del Arduino Pro Micro Fuente: (Sparkfun Electronics, 2016) La fuente de poder de todo el circuito es el Arduino Pro Micro pues toma su energía desde el puerto USB de la computadora y la regula. De aquí se toma la salida de 3,3 voltios para alimentar todos los dispositivos, obteniendo la señal de tierra de referencia (GND).

Debido a que el Arduino Pro Micro requiere un reinicio o "reset" antes de su programación (para ingresar a modo bootloader) se usa un pulsador normalmente abierto conectado entre los pines de Reset y Tierra (GND) para facilitar la programación una vez que el dispositivo esté montado en la placa de circuito impreso. Debido al uso del puerto USB integrado en el Pro Micro no es requerido ningún terminal adicional para conectarlo al computador, lo que sí se requiere es un interruptor para encender y apagar el "modo ratón".

Si dejamos al Arduino en "modo ratón" todo el tiempo luego será muy difícil hacer algún cambio en el programa o mejorar el código, pues siempre aparecerá en la computadora como un ratón. Para esto se usa un pin de entrada digital que en nivel uno lógico (1L) activa el "modo ratón" y en cero lógico (0L) lo desactiva. Esto se logra con un simple divisor de voltaje con una resistencia de 10K Ω conectado entre GND y el interruptor al pin digital 9 del Arduino Pro Micro.

Con todo lo indicado, el diagrama esquemático completo del dispositivo es como se indica en la figura 2.6. / Figura. 2. 6. Diagrama esquemático del prototipo Fuente: Elaborado por el autor En el diagrama esquemático se observa en J1 la conexión al módulo de posicionamiento. Como se puede notar, al ser un dispositivo completamente digital, el diagrama esquemático es muy sencillo.

Como fuente de luz infrarroja se emplea un circuito serie montado a manera de lápiz, el esquemático del circuito está indicado en la figura 2.7: / Figura. 2. 7. Diagrama esquemático del lápiz infrarrojo Fuente: Elaborado por el autor 2.2: ALGORITMO DE PROCESAMIENTO DE LA SEÑAL Para el desarrollo del algoritmo se utilizó el ambiente de diseño integrado (IDE Integrated Development Environment) de Arduino.

Este IDE permite el uso del puerto serie del Arduino Pro Micro para confirmar los datos conforme son recibidos. El algoritmo completo está en el Anexo 2. Para lograr la

comunicación I2C de los dispositivos se emplean las siguientes librerías de Arduino: Wire.h: que incluye todo lo necesario para la comunicación I2C. PVision.h: que incluye la inicialización I2C del módulo y todo lo necesario para recibir correctamente la señal. Mouse.h: que permite el modo de emulación de ratón o "modo ratón". Estas librerías son públicas (están disponibles en <http://www.arduino.cc> (Arduino, 2018)) y permiten su modificación. Se modificó el algoritmo para para obtener los datos como son requeridos.

El algoritmo del programa se construye en "partes", siguiendo la estructura de un programa de Arduino; de la siguiente manera: Parte 1: se declaran las variables y se realiza el direccionamiento de la comunicación Parte 2: se configura la comunicación serie y cómo se indicarán los datos Parte 3: se realiza el procesamiento de la señal recibida y se configura el despliegue de la información Parte 4: se envía la información procesada al "modo ratón" para que la computadora la use El diagrama de flujo que permitió desarrollar el algoritmo se observa en la figura 2.8. / Figura. 2. 8.

Diagrama de flujo del algoritmo Fuente: Elaborado por el autor Dado que el módulo de posicionamiento es capaz de procesar hasta cuatro fuentes de luz infrarroja de manera simultánea, y no son requeridas tantas fuentes de manera simultánea, se deben eliminar las señales no requeridas de manera de realizar el procesamiento únicamente de la primera fuente.

Entonces se envía o "imprime" la información en el puerto serie, o mejor dicho, programar al sistema para que despliegue la información recibida – que corresponde a las coordenadas y al tamaño de la burbuja o "blob" – en el puerto serie del IDE asignado al Arduino Pro Micro. Una vez obtenida la información de posición (las coordenadas X y Y) y confirmado que el sistema está enviando información se programa el funcionamiento del ratón, con un pin que lo habilita y lo deshabilita, para escalar las coordenadas recibidas y mover el ratón en base a eso.

Las coordenadas recibidas varían desde 1023 hasta 0; siendo 1023 la no presencia de luz infrarroja y 0 la última coordenada hacia la derecha en el eje X (o hacia arriba en el eje Y); por lo que, se debe escalar dichas coordenadas a 12 niveles de movimiento; sino es hecho este escalamiento el sistema será muy "sensible" a los movimientos del LED infrarrojo y prácticamente imposible de controlar.

Esta información se debe escalar para una resolución de un monitor de 1024 x 768 pixeles (para el eje Y se divide por 1,33 debido a la relación de la resolución de pantalla 4:3); y, dentro de los rangos de movimiento disminuir el campo de acción del puntero. Otra parte del programa, aprovechando que el módulo es capaz de indicar la presencia

de una burbuja (blob) y su tamaño, hacemos que el Arduino envíe a la computadora la orden de click izquierdo del ratón "mouse-click" de manera de activar la función siempre que el tamaño del blob sea mayor a un valor especificado previamente.

Con esto, únicamente se usa el click izquierdo del ratón, pues para el uso de interactividad no se requieren las funciones de click derecho.

CAPÍTULO 3 DESARROLLO El uso extenso de microcontroladores ha permitido que muchos de los prototipos sean de tamaño muy pequeño, comparados con el tamaño que tendrían los prototipos si se empleasen dispositivos discretos.

Este proyecto no es la excepción, incluso si se usare otro tipo de Arduino (que soporte línea de datos I2C de 3,3v), se podría hacer mucho más pequeño. La idea de hacer el prototipo pequeño fue debido a que normalmente será instalado junto a un proyector por lo que el espacio y peso son muy importantes. El prototipo montado sobre un proyector se observa en la figura 3.9. / Figura. 3. 9. Dispositivo instalado junto al proyector Fuente: Elaborado por el autor

3.1: IMPLEMENTACIÓN

Para obtener la señal de reloj del Arduino Pro Mini fue necesario modificar el bootloader del Arduino.

Así entonces, se obtiene la señal de reloj del sistema en un pin de conexión. Para hacer esto se usó un programador para el ATmega, que puede ser otro Arduino, se modifica el archivo "boards.txt" del Arduino IDE para que indique un tipo "nuevo" de Arduino. Es un tipo "nuevo" pues no es el bootloader estándar de un Arduino.

La conexión para esta programación se observa en la figura 3.10, empleando un Arduino Nano como ISP (In System Programmer o programador en el sistema) para la programación de cualquier otro Arduino (en el Arduino a programar se conectan las señales RST, MOSI, MISO y SCK en los pines correspondientes): / Figura. 3. 10. Diagrama de conexión entre Arduinos para programación de bootloader.

Fuente: (Derwyn, 2017) Una vez realizadas las conexiones apropiadas (en los pines apropiados en base a la señal) se configuró el software Arduino IDE para que emplee un programador ArduinoISP (esto es, debido a que se dispone de uno), se cambió el bootloader modificando el fusible de configuración que indica SCKO o habilitar la salida de reloj (clock out). Para facilitar el cálculo es posible usar una calculadora web (Häm_mer_ling, 2008) (www.engbedded.com/fusecalc) que es una aplicación que calcula los fusibles del ATmega 328P.

Una vez el sistema ejecute la programación, tendremos listo el reloj en el pin de salida del Arduino Pro Mini., como se observa en la figura 3.11. / Figura. 3. 11. Imagen del cálculo del fusible del Arduino Pro Mini Fuente: (Häm_mer_ling, 2008) Para terminar, se debe modificar el archivo boards.txt (el que lista el tipo de tarjetas usadas) y crear una placa Arduino diferente, por ejemplo: "Arduino Pro Mini con reloj" y en la configuración de los fusibles bajos se debe escribir la opción que nos arroja la página de arriba (0xA2) en lugar de la configuración original 0xFF; dicho cambio se observa en negritas en la figura 3.12. Figura. 3. 12.

Configuración de fusibles del ATmega 328P Fuente: Elaborado por el autor Partiendo del diagrama esquemático, y con todos los componentes preparados para la implementación, se procedió a construir la placa de circuito impreso. El método más sencillo para diseño es emplear una aplicación de computadora; pues esta permite automatizar la elección del tipo de línea o pista, revisa los errores u omisiones de conexiones, entre otras funciones.

Entonces, para el diseño se tomó en cuenta algunas consideraciones (el cálculo del ancho de las pistas se basa en el estándar general para diseño ANSI-IPC-2221), las cuales son: Las líneas de circuito impreso que lleven alimentación (dado el bajo consumo de corriente de los circuitos integrados, y que operará a un máximo de 10°C sobre la temperatura ambiente) deben ser de al menos 0,8 mm.

Las líneas de señal de circuito impreso son de muy baja corriente por lo que al menos deben ser de 0,5 mm Esto se hace indicando al sistema de diseño que tipo de línea es, así el software es capaz de elegir el grosor de las líneas y pistas de circuito impreso. El software permite al usuario observar las pistas de circuito impreso desde el lado de los componentes o desde el lado del cobre de la placa, facilitando así el proceso de diseño. Como se observa en la figura 3.13. / Figura. 3. 13.

Placa de circuito impreso vista desde el lado de los componentes Fuente: Elaborado por el autor Con esto, al mandar a imprimir la placa se observa como quedan las "pistas" de cobre, en la figura 3.14. / Figura. 3. 14. Placa de circuito impreso vista desde el lado cobre Fuente: Elaborado por el autor Usando técnicas de fabricación de placas de circuito impreso (transferencia térmica, con marcador, por CNC, etc.) se fabricó la placa. En este caso la placa es de 10 x 10 cm.

De la figura 11 se observa que se usan dos Arduinos, el de la izquierda corresponde al usado como reloj (Arduino Pro Mini) pues solamente tiene una conexión en D8 (y la alimentación eléctrica) y el de la derecha es el procesador de señal (Arduino Pro Micro). El conector J1 es el conector de la parte de arriba de la placa. Debido a que el módulo es miniatura este conector está planificado para "conectar" con cables terminales a dicho módulo, elevándolo sobre la placa para lograr una mejor visión. En la figura 3.15 se observa la placa desde el lado de cobre, con los componentes soldados; y, en la figura 3.16 la placa desde el lado de los componentes: / Figura. 3. 15.

Placa de circuito impreso (lado cobre) con los componentes soldados Fuente: Elaborado por el autor / Figura. 3. 16. Placa de circuito impreso vista desde el lado de los componentes Fuente: Elaborado por el autor Para el lápiz, que es la interfaz de usuario del sistema, y se requiere que sea fácil de manejar como si fuese un marcador estándar;

el camino más sencillo fue emplear el cuerpo de un marcador de pizarra líquida, reemplazando la punta por un diodo LED infrarrojo y usando el cuerpo restante para soportar las baterías. Como se observa en la figura 3.17. / Figura. 3. 17.

Punta del marcador, se observa el interruptor pulsador y el LED IR Fuente: Elaborado por el autor Este diseño permite que la batería (que es la parte más pesada) descansa sobre la palma de la mano al usar el marcador. La idea es que su uso sea lo más natural posible para que la manipulación no se sienta forzada, como indica la figura 3.18. / Figura. 3. 18.

Modo de uso del lápiz IR Fuente: Elaborado por el autor Con todos los componentes construidos es necesario realizar pruebas de operación. Dichas pruebas deben permitir validar la operación del prototipo en un ambiente controlado, se debe confirmar que existe información, que la operación básica deseada se cumpla y que, en general, el dispositivo esté operando. 3.1: PRUEBAS Las pruebas se realizaron cuando todo fue ensamblado.

Consistieron en confirmar que el módulo esté enviando señales de posicionamiento y que la señal sea apropiada para el uso en el computador. Esto se logra abriendo el puerto serie en el Arduino IDE (Ctrl+Shift+M), el sistema – en ausencia de luz infrarroja – mostrará lo indicado en la figura 3.19. / Figura. 3. 19.

Ventana del puerto serie en ausencia de luz infrarroja Fuente: Elaborado por el autor Si se presenta una fuente infrarroja el sistema debe indicar las coordenadas relativas de la posición, en base a su ubicación, como se ve en la figura 3.20. / Figura. 3. 20. Ventana de posición con una fuente de luz infrarroja Fuente: Elaborado por el autor Con la información presentada en la figura 3.20 se puede comprobar que el módulo está enviando la información y es procesada de manera adecuada.

Otra prueba de operación se realizó encendiendo la funcionalidad de ratón (mouse) y observando el comportamiento en la computadora. Se observa que al mover el lápiz infrarrojo el cursor del ratón de la computadora respondió a los movimientos. Ahora bien, existe una aplicación que permite también procesar la información enviada por el dispositivo en relación con el tamaño del blob y su posición, esta aplicación se llama Processing y es empleada por desarrolladores de todo el mundo para procesar resultados de aplicaciones diseñadas.

En esta ventana se puede observar el comportamiento del dispositivo sin mover el ratón del sistema operativo, como se aprecia en la figura 3.21. / Figura. 3. 21. Ventana de Processing Indica de manera gráfica la posición del lápiz infrarrojo (mancha blanca en la

pantalla), indicando las coordenadas en la parte inferior. Fuente: Elaborado por el autor

3.2: ANÁLISIS DE RESULTADOS

En base a las dos pruebas realizadas, los resultados se observan en la tabla 3.5.: Tabla. 3. 5.

Tabla comparativa de resultado Prueba

Prueba	Resultado
En ausencia de luz infrarroja	El dispositivo emite información nula de coordenadas
El dispositivo, en ausencia de luz infrarroja	emite coordenadas 0,0
En presencia de luz infrarroja	El dispositivo emite información de coordenadas
El dispositivo, en presencia de luz infrarroja	emite coordenadas relativas
En presencia de luz	El dispositivo es sensible
El dispositivo presenta una operación	extremadamente sensible, el menor movimiento de la luz hace que el resultado de la operación se mueva demasiado rápido
Observar el comportamiento del dispositivo	en estado de reposo
Se nota que en reposo (sin fuente de luz)	a momentos el dispositivo presenta blobs que son causados como ruido por la luz natural
Activar la operación de "modo ratón"	y observar comportamiento
Con el "modo ratón" activado	el movimiento de la fuente de luz hace que el cursor del ratón del sistema operativo se mueva en concordancia con el movimiento de la fuente de luz
Con "modo ratón" activo	el cursor debe responder al movimiento de la fuente de luz
Se observa que el sistema responde,	aunque de manera muy sensible.

__Fuente: Elaborado por el autor

Prueba 1: Si no existe ninguna luz infrarroja el dispositivo envía coordenadas 0,0; indicando ausencia del lápiz infrarrojo.

Al presentar el lápiz el dispositivo indica su posición relativa dentro del campo de visión de este enviando coordenadas, se observó que el comportamiento del dispositivo es extremadamente sensible, por lo que fue necesario aumentar un retardo en el programa de manera de poder observar de mejor manera su operación. A momentos el sistema indica coordenadas y blobs sin presencia del lápiz infrarrojo, lo cual es una interferencia no deseada, esto sucede durante el día en sitios con ventanas o cierto tipo de iluminación, la solución para esto es instalar un "filtro infrarrojo" frente al módulo.

Este filtro no es más que un plástico de color violeta, similar al usado en muchos controles remotos, que actúa como filtro a ciertas luces, eliminando por completo la interferencia.

Prueba 2: El sistema es capaz de controlar al ratón del sistema operativo, aunque a momentos es muy sensible, un movimiento leve del lápiz infrarrojo mueve rápidamente al ratón del sistema operativo, esto se corrigió aumentando un retraso en el programa de procesamiento de la señal.

CONCLUSIONES El resultado de este proyecto demuestra que es posible crear dispositivos electrónicos de bajo costo para solucionar requerimientos del mundo real, fuera de pruebas de laboratorio e incluso con capacidad de hacer un desarrollo comercial. El dispositivo cumple con la función de brindar interacción con un proyector, brindando además otras posibles funciones (para futuros desarrollos) como: interacción con PC "touch", capacidad de implementar múltiples ratones al tiempo en el sistema operativo entre otras. Con esto se ha cumplido el objetivo general del proyecto.

El uso de componentes de electrónica digital (dispositivos programables) permitió que la creación del dispositivo sea cumplida completamente, permitiendo mejoras futuras y desarrollo de la idea. Esto cumple con el objetivo de definir los elementos constitutivos de un dispositivo electrónico para transformar un proyector estándar en interactivo.

Debido al uso de dispositivos programables el uso de buenas técnicas de programación y diseño (diagrama de bloques, diagrama de flujo, y similares) permitió el desarrollo de la idea de manera práctica, con un desarrollo funcional casi inmediato. Cumpliendo así el objetivo de elaboración del diagrama de bloques. Partiendo de los pasos ordenados indicados arriba se facilitó el desarrollo del diagrama esquemático del circuito.

Realizando un análisis del flujo y tratamiento de la señal el proceso de diseño del circuito electrónico se transforma en un proceso intuitivo, permitiendo así, cumplir con el objetivo trazado. De igual manera, una vez planificado diagrama de bloques, flujo, esquemático y definido el procesador a usar la creación o modificación del algoritmo de programación se transformó en una tarea no compleja.

Con todos los pasos cumplidos (diseño del circuito electrónico y algoritmo de programación) se procedió a implementar el dispositivo. El definir un esquema de pruebas permitió que se valide la operación deseada del dispositivo, facilita el proceso de corrección de errores y permite una mejora continua del mismo (de ser deseado) para futuros desarrollos.

RECOMENDACIONES En el mercado existen varios tipos de microcontroladores; para una implementación sencilla de tipo experimental, como la planteada en el presente proyecto, el uso de dispositivos que puedan ser programados sin necesidad de removerlos de la placa de circuito impreso facilita mucho el trabajo, pues remover el microcontrolador de la placa electrónica en las pruebas puede generar daños además de generar retrasos, gastos no planificados y cuestiones similares.

En este proyecto se emplea un protocolo de comunicación propietario entre los circuitos integrados (I2C), el cual tiene ciertas características que complican un poco el diseño; por ejemplo, el hecho de que la línea de datos sea de 3,3 voltios limita bastante la elección del microcontrolador, pues muchos de los dispositivos disponibles comercialmente emplean una línea de datos de 5 voltios.

Esto puede causar que sean requeridos dispositivos adicionales para el ajuste de señal, lo cual complica el diseño, aumenta los costos y disminuye la confiabilidad. Como todo dispositivo electrónico se debe validar su operación mediante pruebas de funcionamiento; con las cuales, si el dispositivo no funciona como se espera, se puede corregir errores para lograr la funcionalidad deseada.

Previo a iniciar el proceso de diseño, cuando se están analizando opciones de dispositivos a emplear, se debe analizar a detalle las funciones y características de los dispositivos elegidos; pues al ser programables, el diseñador puede encontrar situaciones en las cuales éstos no funcionen de la manera esperada, tengan falta de funciones o no sean adecuados para el uso planificado.

Esto generará frustración en el diseñador, aumento de costos, plazos o que simplemente el prototipo no sea viable.

BIBLIOGRAFÍA Arduino. (2018). Arduino - Home. Obtenido de <https://www.arduino.cc/en/Reference/> Derwyn. (04 de Junio de 2017). Arduino NANO as ISP (programmer). Obtenido de <https://gw4sae.wordpress.com/2017/06/04/arduino-nano-as-isp-programmer/> Häm_mer_ling, M. (01 de June de 2008). Engbedded AVR Fuse Calculator. Obtenido de <http://www.engbedded.com/fusecalc> Hobley, S. (22 de Febrero de 2009). www.stephenhobley.com.

Obtenido de www.stephenhobley.com: <http://www.stephenhobley.com> Lee, J. C. (2 de Noviembre de 2016). Procastineering - Project blog. Obtenido de <http://procastineering.blogspot.com/> Microsoft. (09 de Septiembre de 2011). Interpreting Mouse Axis Data. Obtenido de [https://docs.microsoft.com/en-us/previous-versions/windows/desktop/ee418272\(v=vs.85\)](https://docs.microsoft.com/en-us/previous-versions/windows/desktop/ee418272(v=vs.85)) Simple Machines. (2017). Wiimote Project. Obtenido de <https://www.wiimoteproject.com/index.php> Sparkfun Electronics. (13 de Marzo de 2016).

Using the Arduino Pro Mini. Obtenido de Learn Sparkfun: <https://learn.sparkfun.com/tutorials/using-the-arduino-pro-mini-33v> Wikipedia. (2006). Wii Remote. Obtenido de https://en.wikipedia.org/wiki/Wii_Remote

INTERNET SOURCES:

0% - Empty