Universidad de Costa Rica

Facultad de Ingeniería
Escuela de Ingeniería Eléctrica
IE0624 – Laboratorio de Microcontroladores
III ciclo 2022

Reporte 2

GPIOs, Timers y FSM

Ana Eugenia Sánchez Villalobos B87382
Adrián Montero Bonilla B85092
Grupo 01

Profesor: Marco Villalta

23/01/2023

Índice

Índice de figuras

1. Resumen

En este Laboratorio 2 se desarrolla el juego de memoria llamado Simon dice, donde por medio de entradas y salidas GPIOS, interrupciones, máquinas de estados finitos y ATtiny4313, se logra una interacción con el usuario mediante el juego de memoria. Se plantea el juego Simon dice, mediante el uso de 4 LEDs que iluminan indicando un patrón que el usuario debe acertar. En caso de no acertar, este vuelve inicialmente a su nivel más bajo, y lo comunica mediante el parpadeo de las luces LED tres veces, o en caso contrario sube de nivel a uno más difícil. La secuencia tiene como máximo un total de 14 LEDs, por tanto este sería el nivel más alto que se podría tener en el juego. Dentro del juego, se toma en cuenta el tiempo que duran los LEDs encendidos, mediante el uso de temporizadores del microcontrolador, conforme se sube de nivel deben ir bajando 200 microsegundos de tiempo.

2. Nota Teórica

A continuación se detalla la información del microcontrolador `ATtiny4313`, así como los periféricos utilizados, componentes electrónicos y otra información relevante.

2.1. Caraterísticas generales del AT-tiny2313

El microcontrolador AT-tiny2313, pertenece a la familia de microcontroladores AVR, desarrollado desde 1996. Este microcontrolador incluye arquitectura Harvard modificada, en otras palabras arquitectura RISC avanzada. Cuenta con un reloj interno de 20MHz y 20 pines en total, de los cuales 18 son programables como entradas o salidas (GPIOS), estos están divididos en tres puertos y cada uno de ellos cumple con una característica especial, y los otros dos corresponden a tierra y a source. En la figura 1, se puede observar el diagrama de pines de este microcontrolador.

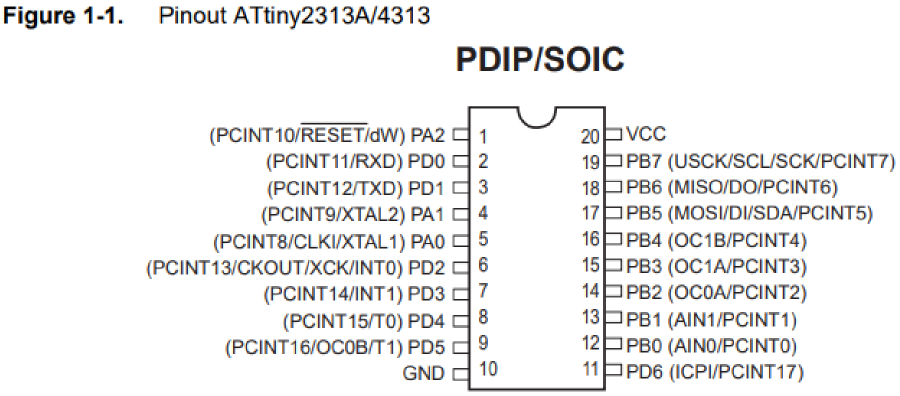


Figura 1: Diagrama de pines del AT-tiny2313 [1]

Las características generales del microcontrolador, se pueden observar en la siguiente tabla:

Microcontrolador AT-tiny2313	
Tipo de programa de memoria	Flash
Data EEPROM (bytes)	128
Tensión de Operación (min - máx)	1.8V - 5.5V
Velocidad CPU (MIPS)	20
Timers	1x8bits - 1x 16bits
Rango de temperatura (min - máx)	-40°C -85°C
Canales ADC	0

Tabla 1: Características generales del AT-tiny2313. [1]

Por el otro lado para comprender más el comportamiento del microcontrolador, podemos observar su diagrama de bloques en la figura 9, que nos ayuda a comprender mejor el comportamiento de las memorias, los registros y temporizadores.

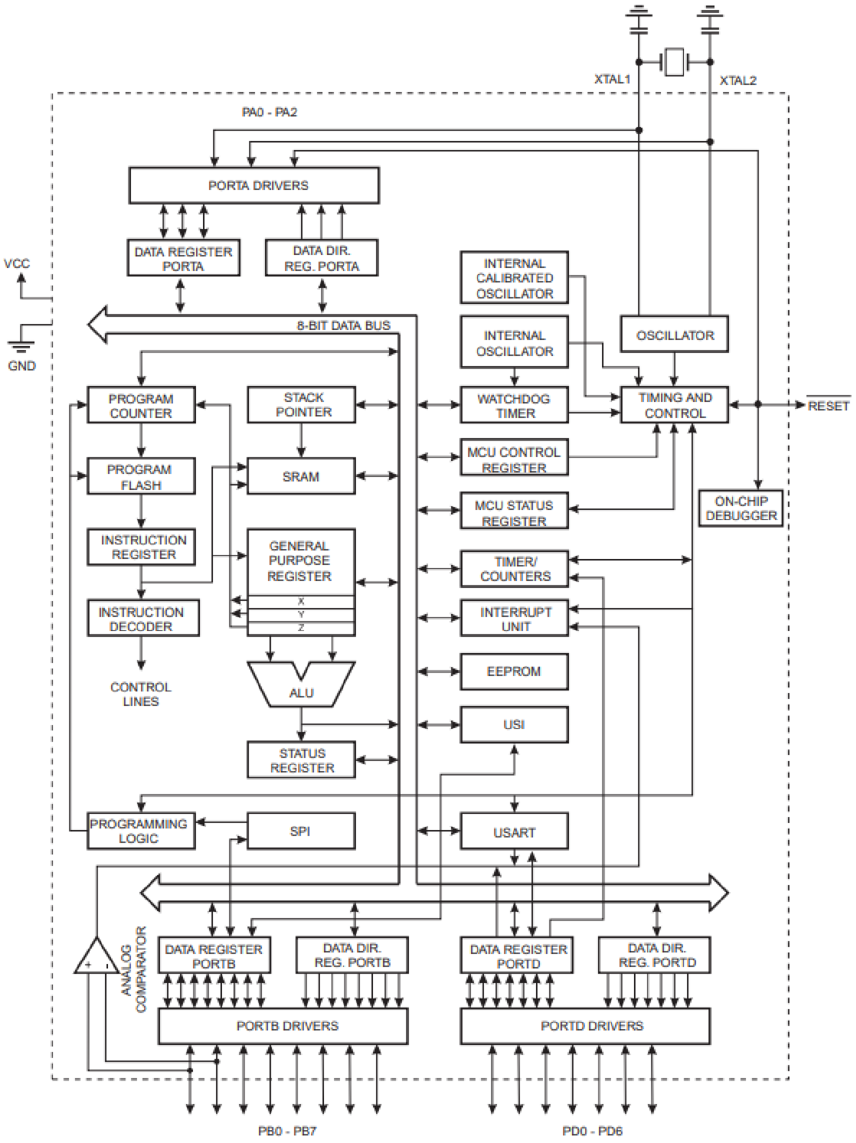
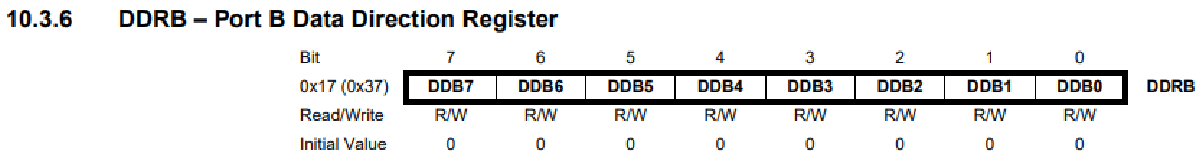


Figura 2: Diagrama de bloques del AT-tiny2313 [1]

2.2. Periféricos utilizados

2.2.1. Registro DDRB

Este es el registro de configuración de los pines del puerto B, también conocido como PORTB. Con este registro se le puede decir la microcontrolador cuales pines van a ser tomados como ïnputs". La configuración general del registro se puede observar a continuación:



10.3.6 DDRB – Port B Data Direction Register

Bit	7	6	5	4	3	2	1	0	
0x17 (0x37)	DDB7	DDB6	DDB5	DDB4	DDB3	DDB2	DDB1	DDB0	DDRB
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Figura 3: Configuración general del registro DDRB

Un ejemplo del registro utilizado en el programa de Simon dice, se puede observar en el código 2. En esta parte del código se puede observar la función `setup()`, que se encarga de configurar el inicio del juego. En la línea 2, del código 2, podemos observar que se configura la entrada y salida de los puertos B, en este caso se indica los puertos en donde van a ir conectados los LEDS, que son salidas. El valor predeterminado de ellos va a ser 0, el cual indica que es una entrada, o en el caso contrario se puede modificar el registro con 1, en el pin que queremos como salida. En caso de omitir este registro se toman todos los pines como su valor predeterminado. [1]

```
1  void setup(){
2    DDRB = 0x0F; // Configuracion del puerto B, 0 es input y 1 es output
3    GIMSK = 0xE8; // Habilitamos interrupciones INT0, INT1, PCIE0 y PCIE2
4    PCMSK = 0b10000000; // Solo el pin con PCINT7 (PB7) puede disparar la interrupcion PCIE0.
5    PCMSK1 = 0b00000100; // Solo el pin con PCINT10 (PA2) puede disparar la interrupcion PCIE1.
6    MCUCR = 0x0A; // Solo los flancos negativos en INT0-1 generan una interrupcion.
7    state = IDLE;
8    playing = 0, button = 0, counter = 0, seed_counter = 0, indicador = 1; turn = 1;
9    sei(); // Habilitamos dentro de C las ISRs.
10   srand(seed_counter); // Sirve para crear una semilla para los n meros random.
11   reset_arrays();
12 }
```

Código 1: Función `setup()`

2.2.2. Registro Puerto B

El puerto B se asocia indirectamente con el registro DDRB, ya que ambos dictan comportamientos de los pines B. En este caso este se encarga de decirle al microcontrolador si el pin B designado en el registro DDRB se encuentra encendido o apagado. Este es bidireccional y sus pines tienen tres posibles estados, en el cual cuando entra en un estado de reset, la condición se activa, esto incluso cuando el reloj no este en funcionamiento. En la siguiente imagen se puede observar una tabla que dicta las funciones alternativas que tiene cada pin. [1]

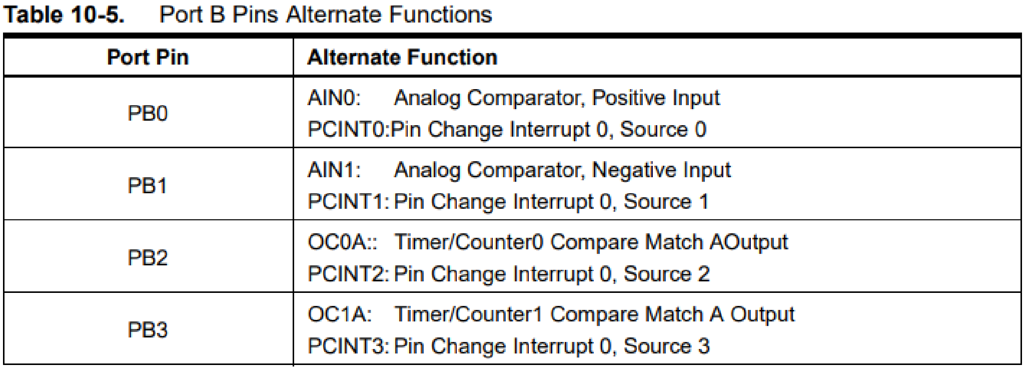


Table 10-5. Port B Pins Alternate Functions

Port Pin	Alternate Function
PB0	AIN0: Analog Comparator, Positive Input PCINT0:Pin Change Interrupt 0, Source 0
PB1	AIN1: Analog Comparator, Negative Input PCINT1: Pin Change Interrupt 0, Source 1
PB2	OC0A:: Timer/Counter0 Compare Match AOutput PCINT2: Pin Change Interrupt 0, Source 2
PB3	OC1A: Timer/Counter1 Compare Match A Output PCINT3: Pin Change Interrupt 0, Source 3

Figura 4: Configuración general del registro DDRB

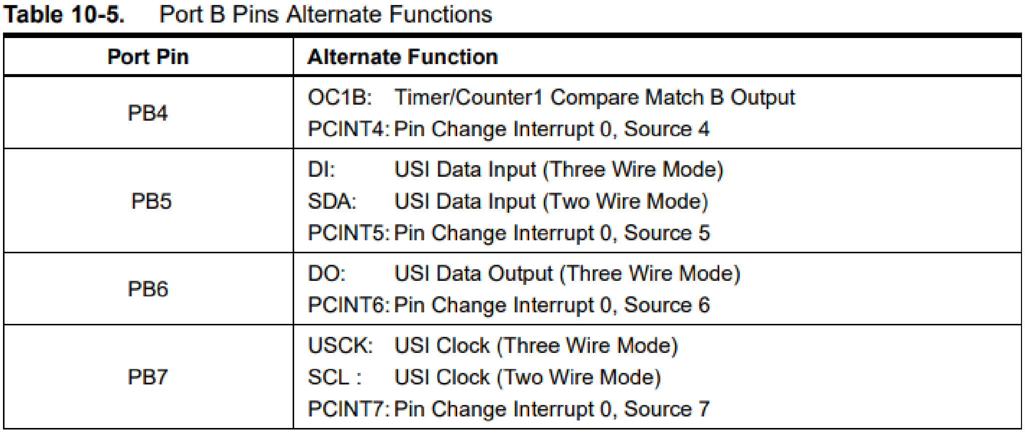


Table 10-5. Port B Pins Alternate Functions

Port Pin	Alternate Function
PB4	OC1B: Timer/Counter1 Compare Match B Output PCINT4: Pin Change Interrupt 0, Source 4
PB5	DI: USI Data Input (Three Wire Mode) SDA: USI Data Input (Two Wire Mode) PCINT5: Pin Change Interrupt 0, Source 5
PB6	DO: USI Data Output (Three Wire Mode) PCINT6: Pin Change Interrupt 0, Source 6
PB7	USCK: USI Clock (Three Wire Mode) SCL : USI Clock (Two Wire Mode) PCINT7: Pin Change Interrupt 0, Source 7

Figura 5: Configuración general del registro DDRB

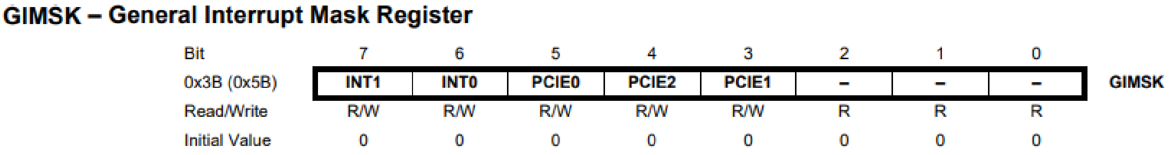
Un ejemplo de cómo se pueden encender o apagar los pines se muestra en el siguiente código:

```
PORTB = 0b00000100;
```

Código 2: Ejemplo de puerto PORTB

2.2.3. Registro GIMSK

El registro GIMSK, esta asociado con las interrupciones, que son los eventos que detienen el flujo de instrucciones en el procesador, y ejecutan una tarea. Al terminar esta tarea, el procesador sigue su curso en el punto en donde fue interrumpido. Con el uso de interrupciones es posible reconocer un botón que nos permita la interacción con el usuario. La interrupciones pueden ocurrir cuando hayan un cambio de nivel, esto también incluyen flancos positivos o negativos. La configuración de este registro se puede observar en la figura 6. [1]



GIMSK – General Interrupt Mask Register

Bit	7	6	5	4	3	2	1	0	
0x3B (0x5B)	INT1	INT0	PCIE0	PCIE2	PCIE1	–	–	–	GIMSK
Read/Write	R/W	R/W	R/W	R/W	R/W	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	

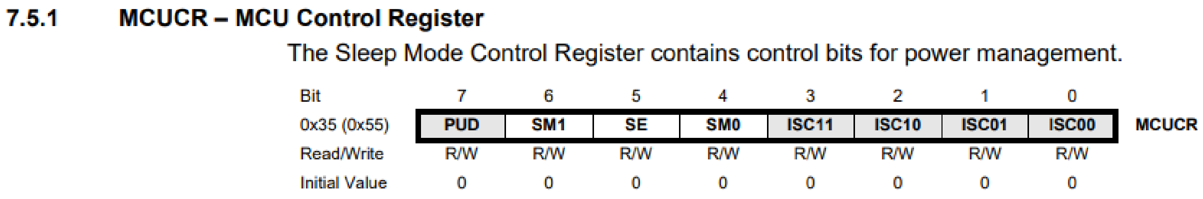
Figura 6: Configuración general del registro GIMSK

Como se puede observar los bits del 0 al 2 siempre se van a leer como cero. Por el otro lado del 3 al 7 pueden ser modificados. Puertos como el INT1 son los que pueden ser utilizados como interrupciones externas.

2.2.4. Registro MCUCR

Este registro controla las interrupciones externas, en este caso se revisa las interrupciones cuando haya un cambio de nivel, influyendo flanco positivo o negativo. En este caso se configura para que cuando detecte un flanco negativo (soltamos el botón), se genere una interrupción en INT0-1. En el código 2, podemos observar en la línea 7, la configuración de este registro. [1]

A continuación se muestra la configuración general del registro:



7.5.1 MCUCR – MCU Control Register

The Sleep Mode Control Register contains control bits for power management.

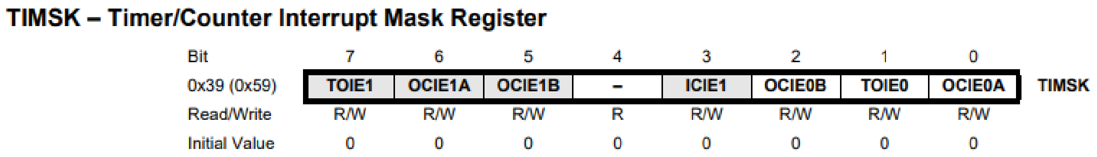
Bit	7	6	5	4	3	2	1	0	
0x35 (0x55)	PUD	SM1	SE	SM0	ISC11	ISC10	ISC01	ISC00	MCUCR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Figura 7: Configuración general del registro MCUCR

Este registro tiene la particularidad que los bits de 6 al 4, pueden configurar el Sleep Mode, poniendo este en Idle, Power-down, Standby o Power-down. Como todos los registros, tiene valor predeterminado de cero, a menos de que sea modificado.

2.2.5. Temporizadores

Este microcontrolador se compone de dos contadores para poder cuantificar el tiempo, estos dos registros en caso de no ser modificados, su valor por defecto al inicializarlos es de cero. Los dos registros importantes son los siguientes: TCCR0B y TIMSK. El TIMSK, activa las interrupciones en el contador, y el otro TCCR0B se habilita cuando hay una interrupción. [1] La configuración general se observa en las siguientes figuras:



TIMSK – Timer/Counter Interrupt Mask Register

Bit	7	6	5	4	3	2	1	0	
0x39 (0x59)	TOIE1	OCIE1A	OCIE1B	–	ICIE1	OCIE0B	TOIE0	OCIE0A	TIMSK
Read/Write	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Figura 8: Configuración general del registro TIMSK

2.2.6. Botón push

Con el fin de poder hacer que funcione el botón y que se active la interrupción, se configura una resistencia pull-down que podrá leer una señal en alto por el pin conectado.

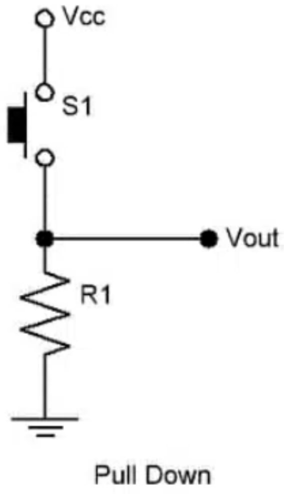


Figura 9: Resistencia Pull-Down. [2]

Este logra generar un paso o no de corriente a los pines diseñados como entrada con los registros DDRB, que en este caso sería nuestro pin de entrada donde entra la tensión. Al utilizar el botón push debemos de tener en cuenta ciertos fenómenos como lo es el efecto rebote. Este efecto produce un fluctuación en la señal poniendo la señal en un estado de alto y bajo, que no queremos que ocurra, esto puede llegar a provocar efectos indeseados o extraños. Para poder evitar este tipo de efectos se pueden solucionar agregando un capacitor antirrebote, en paralelo con la resistencia de pull down. [2]

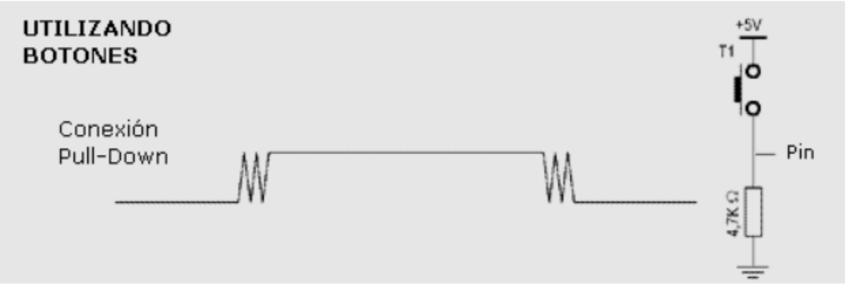
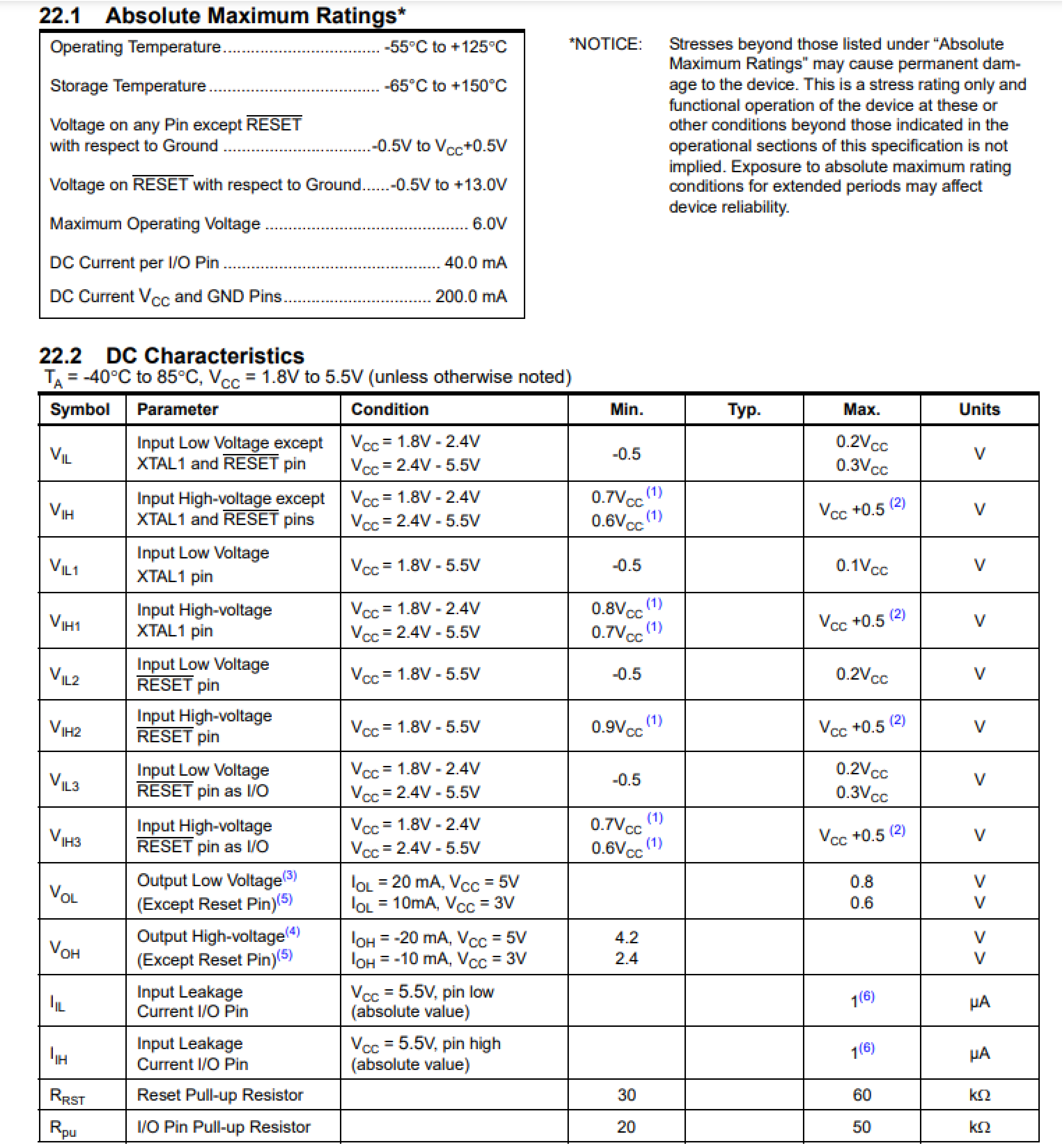


Figura 10: Efecto rebote. [2]

2.3. Características eléctricas del AT-tiny4313



22.1 Absolute Maximum Ratings*

Operating Temperature	-55°C to +125°C
Storage Temperature	-65°C to +150°C
Voltage on any Pin except $\overline{\text{RESET}}$ with respect to Ground	-0.5V to V_{CC}+0.5V
Voltage on $\overline{\text{RESET}}$ with respect to Ground	-0.5V to +13.0V
Maximum Operating Voltage	6.0V
DC Current per I/O Pin	40.0 mA
DC Current V_{CC} and GND Pins	200.0 mA

*NOTICE: Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or other conditions beyond those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

22.2 DC Characteristics

T_A = -40°C to 85°C, V_{CC} = 1.8V to 5.5V (unless otherwise noted)

Symbol	Parameter	Condition	Min.	Typ.	Max.	Units
V_{IL}	Input Low Voltage except XTAL1 and $\overline{\text{RESET}}$ pin	V_{CC} = 1.8V - 2.4V V_{CC} = 2.4V - 5.5V	-0.5		0.2V_{CC} 0.3V_{CC}	V
V_{IH}	Input High-voltage except XTAL1 and $\overline{\text{RESET}}$ pins	V_{CC} = 1.8V - 2.4V V_{CC} = 2.4V - 5.5V	0.7V_{CC} (1) 0.6V_{CC} (1)		V_{CC} +0.5 (2)	V
V_{IL1}	Input Low Voltage XTAL1 pin	V_{CC} = 1.8V - 5.5V	-0.5		0.1V_{CC}	V
V_{IH1}	Input High-voltage XTAL1 pin	V_{CC} = 1.8V - 2.4V V_{CC} = 2.4V - 5.5V	0.8V_{CC} (1) 0.7V_{CC} (1)		V_{CC} +0.5 (2)	V
V_{IL2}	Input Low Voltage $\overline{\text{RESET}}$ pin	V_{CC} = 1.8V - 5.5V	-0.5		0.2V_{CC}	V
V_{IH2}	Input High-voltage $\overline{\text{RESET}}$ pin	V_{CC} = 1.8V - 5.5V	0.9V_{CC} (1)		V_{CC} +0.5 (2)	V
V_{IL3}	Input Low Voltage $\overline{\text{RESET}}$ pin as I/O	V_{CC} = 1.8V - 2.4V V_{CC} = 2.4V - 5.5V	-0.5		0.2V_{CC} 0.3V_{CC}	V
V_{IH3}	Input High-voltage $\overline{\text{RESET}}$ pin as I/O	V_{CC} = 1.8V - 2.4V V_{CC} = 2.4V - 5.5V	0.7V_{CC} (1) 0.6V_{CC} (1)		V_{CC} +0.5 (2)	V
V_{OL}	Output Low Voltage(3) (Except Reset Pin)(5)	I_{OL} = 20 mA, V_{CC} = 5V I_{OL} = 10mA, V_{CC} = 3V			0.8 0.6	V V
V_{OH}	Output High-voltage(4) (Except Reset Pin)(5)	I_{OH} = -20 mA, V_{CC} = 5V I_{OH} = -10 mA, V_{CC} = 3V	4.2 2.4			V V
I_{IL}	Input Leakage Current I/O Pin	V_{CC} = 5.5V, pin low (absolute value)			1(6)	µA
I_{IH}	Input Leakage Current I/O Pin	V_{CC} = 5.5V, pin high (absolute value)			1(6)	µA
R_{RST}	Reset Pull-up Resistor		30		60	kΩ
R_{pu}	I/O Pin Pull-up Resistor		20		50	kΩ

Figura 11: Características eléctricas del microcontrolador ATTiny-4313 [1].

2.3.1. Componentes para tratar el efecto rebote del botón de entrada.

Considerando el efecto rebote que puede existir al pulsar el botón, se diseña un capacitor con un tiempo de descarga y carga de unos cuantos microsegundos, de forma que las pequeñas fluctuaciones de los estados lógicos de los pines causados por el efecto rebote sean filtrados y se tenga una transición suave y sin picos grandes. En la siguiente imagen se puede observar el filtro que elimina el efecto rebote del botón push.

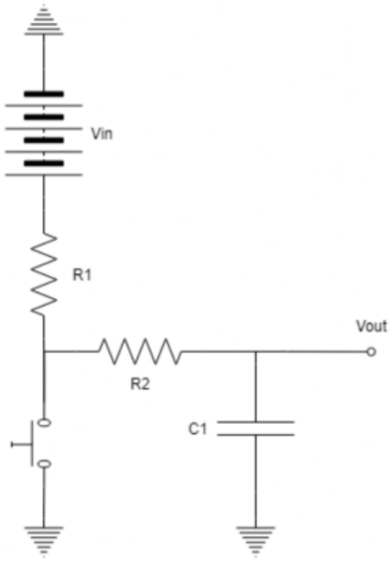


Figura 12: Filtro que elimina el efecto rebote del botón push

Por lo tanto se elige un capacitor comercial de 100nF y una resistencia de 100Ω. Donde si sacamos el tiempo τ, obtenemos que.

$$\tau = RC = 100\Omega * 100nF = 10\mu s \tag{1}$$

Este tiempo es adecuado, ya que no es muy pequeño, pero tampoco muy grande. Este tiempo es captado por el usuario y al mismo tiempo no tiene reacción directa en los tiempos del cambio de luces, los cuales ya están previamente definidos. Con esto los costos son los siguientes:

Componente	**Cantidad**	**Precio en Colones**
ATtiny2313	1	₡ 710
Resistencia 100	8	₡ 1136
LED	4	₡ 1988
Botón push	4	₡ 1136
Capacitor	4	₡ 994

Tabla 2: Valor de componentes. [3]

Donde por ahora el monto total es de ₡ 5964 colones.

2.3.2. Resistores de protección para los LEDs

El diseño de los resistores escogidos vienen de un análisis circuital con las leyes de Kirchhoff. Se escogieron LEDs con umbrales de 1.65V y se aplicó un LTK; para una corriente de 20 mA (valor típico especificado en la hoja del fabricante de los LEDs escogidos) y con una tensión de pin de 4V (se le resta un Volt para considerar pérdidas internas), se tiene la siguiente ecuación:

$$-4 + V_{Rext} + 1{,}65 = 0 \tag{2}$$

Al cambiar V_{Rext} con ley de Ohm y despejar para encontrar el valor de R_{ext} se consigue lo siguiente:

$$
\begin{aligned}
R_{ext} &= \frac{4 - 1{,}65}{20 \cdot 10^{-3}} && (3)\\
&= 118\Omega && (4)
\end{aligned}
$$

3. Desarrollo y Análisis de Resultados

3.1. Diagrama de flujo del juego Simon Dice

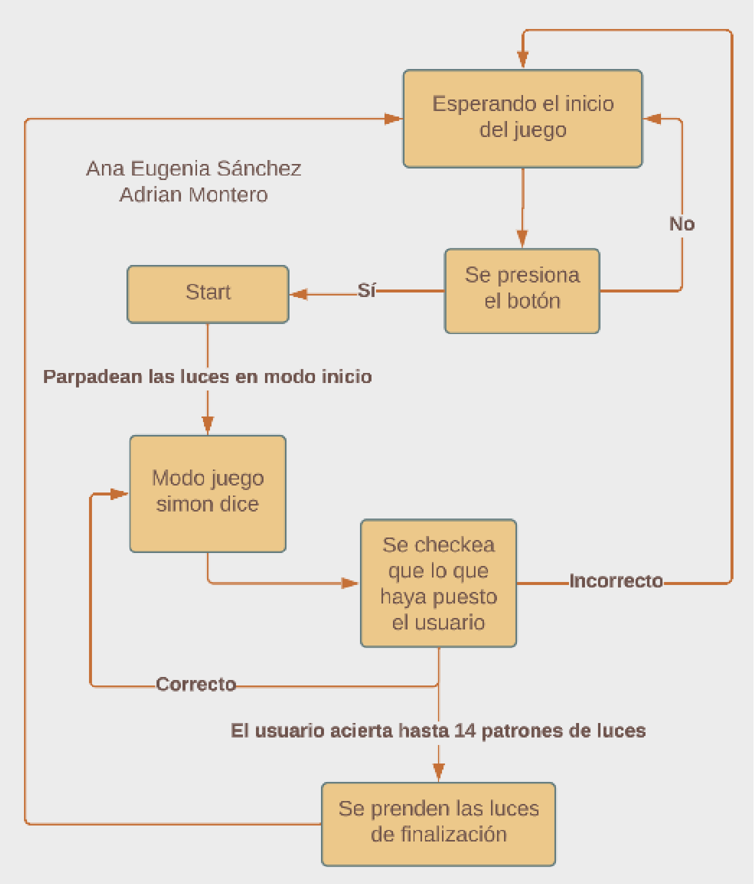

Esperando el inicio del juego
Ana Eugenia Sánchez
Adrian Montero
No
Start
Sí
Se presiona el botón
Parpadean las luces en modo inicio
Modo juego simon dice
Se checkea que lo que haya puesto el usuario
Incorrecto
Correcto
El usuario acierta hasta 14 patrones de luces
Se prenden las luces de finalización

Figura 13: Diagrama de flujo del comportamiento del juego Simon Dice.

3.2. Luces de inicio y del final

Para indicar el inicio y el final del juego, se transmite una señal con las luces. Al inicio del juego al presionar cualquier botón se le indica al usuario el inicio mediante el parpadeo de los LEDs 2 veces. Por el otro lado cuando el juego finaliza ya sea porque el usuario logró llegar al máximo nivel, o por que haya perdido, todos los LEDs parpadean 3 veces.

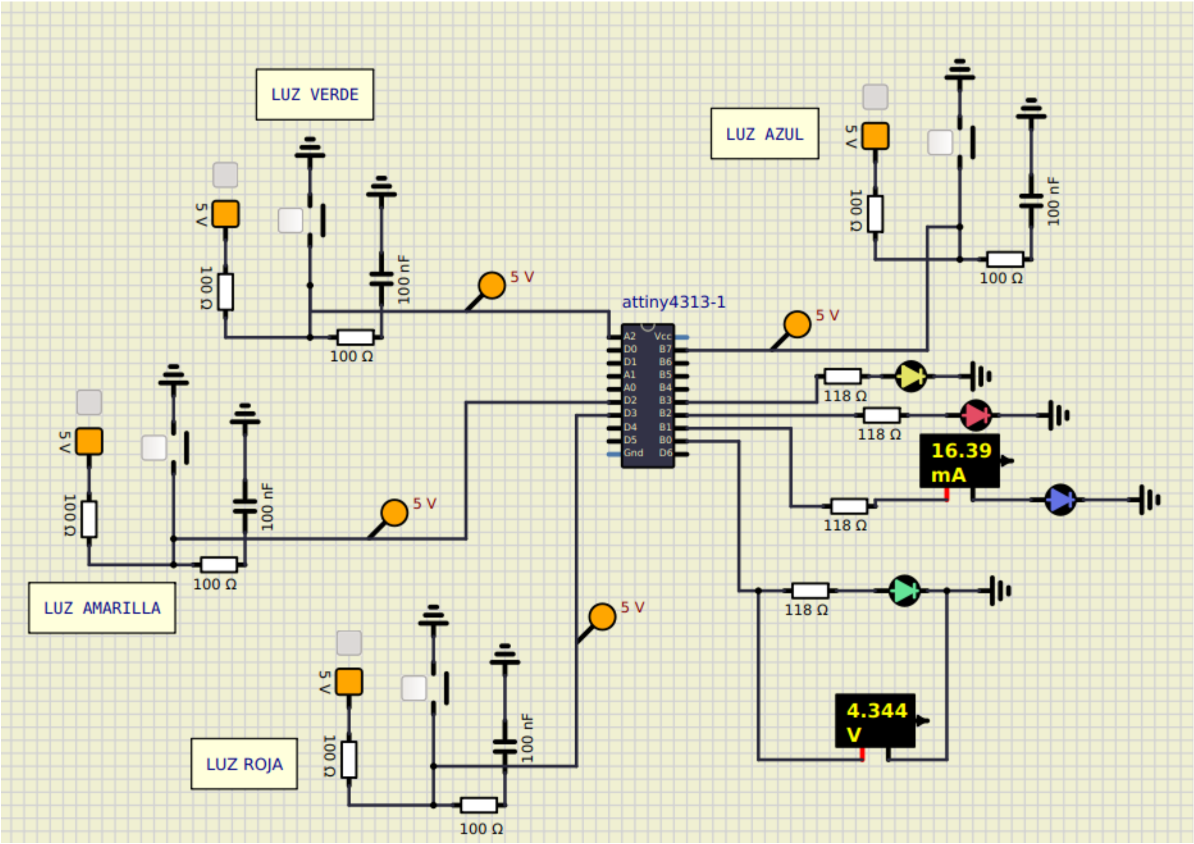


Figura 14: Demostración de luces encendidas.

El comportamiento como tal no se puede demostrar por medio de una fotografía, pero en la figura 14, se puede observar todas las luces encendidas luego de presionar cualquier botón. En este diagrama de la figura 14, podemos destacar los estados de la máquina de estados finita. Donde los estados son:

- **IDLE** : En este estado se esta esperando el inicio del juego, o mejor dicho a que el usuario presione el botón.
- **START** : En este estado ya se presiono el botón y parpadean las luces dos veces para indicarle al usuario que el juego ya ha iniciado.
- **PLAYING** : En este estado comienza el juego, y los patrones de luces comienzan a incrementar conforme el usuario acerta el patrón anterior.
- **CHECK** : Este estado se encarga de verificar que la entrada que introdujo el usuario mediante los botones sea igual al patrón. En caso de no ser igual se dirige al estado de RESET. O por caso contrario de que acierte vuelve nuevamente al estado PLAYING y aumenta el patrón de LEDs.
- **RESET** : En este estado el usuario no ha acertado el patrón por lo que se encarga de resetear los arreglos y las variables necesarias.

En la siguiente tabla se puede observar la transición de estados.

Estado Actual	Estado Siguiente	
	Botón en 0	Botón en 1
IDLE	IDLE	START
START	START	PLAYING
PLAYING	PLAYING	CHECK
CHECK	CHECK	PLAYING o RESET
RESET	RESET	PLAYING o IDLE

Tabla 3: Tabla de Transición de Estados

3.3. Demostración de funcionamiento de las cuatro luces

En las siguientes figuras, podemos observar el funcionamiento de cada LED por separado, por motivos obvio no es posible demostrar los patrones. Por otro lado es importante mencionar que se colocan probes, un amperímetro y un voltímetro, para verificar el buen funcionamiento del circuito.

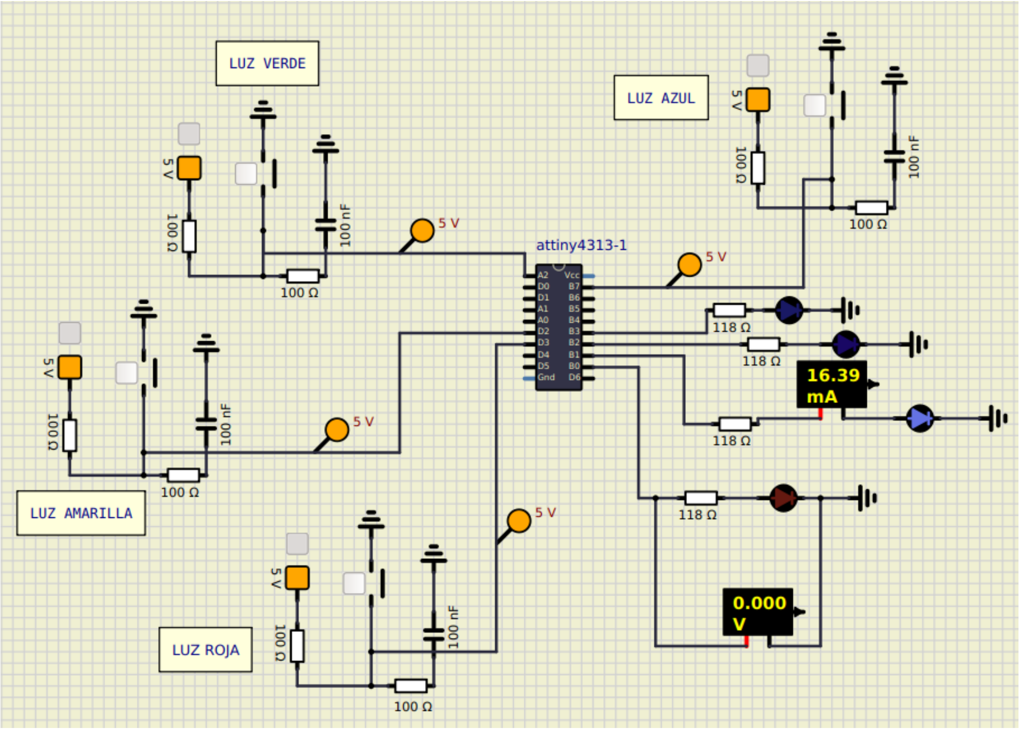


Figura 15: Funcionamiento de LED azul.

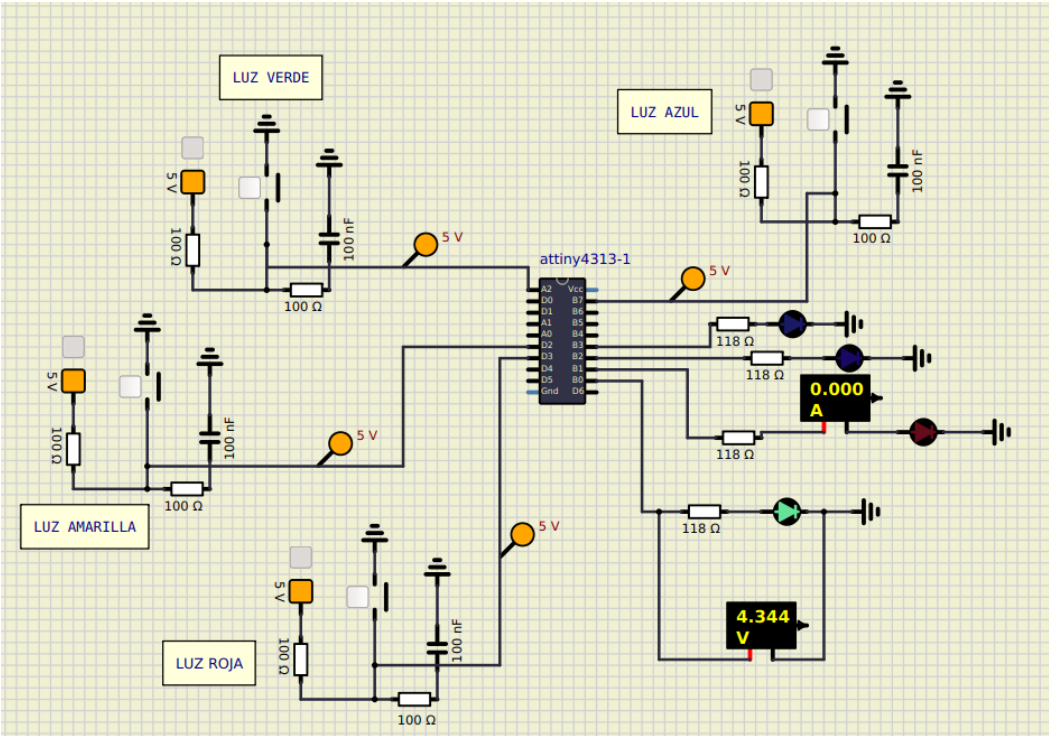


Figura 16: Funcionamiento de LED verde.

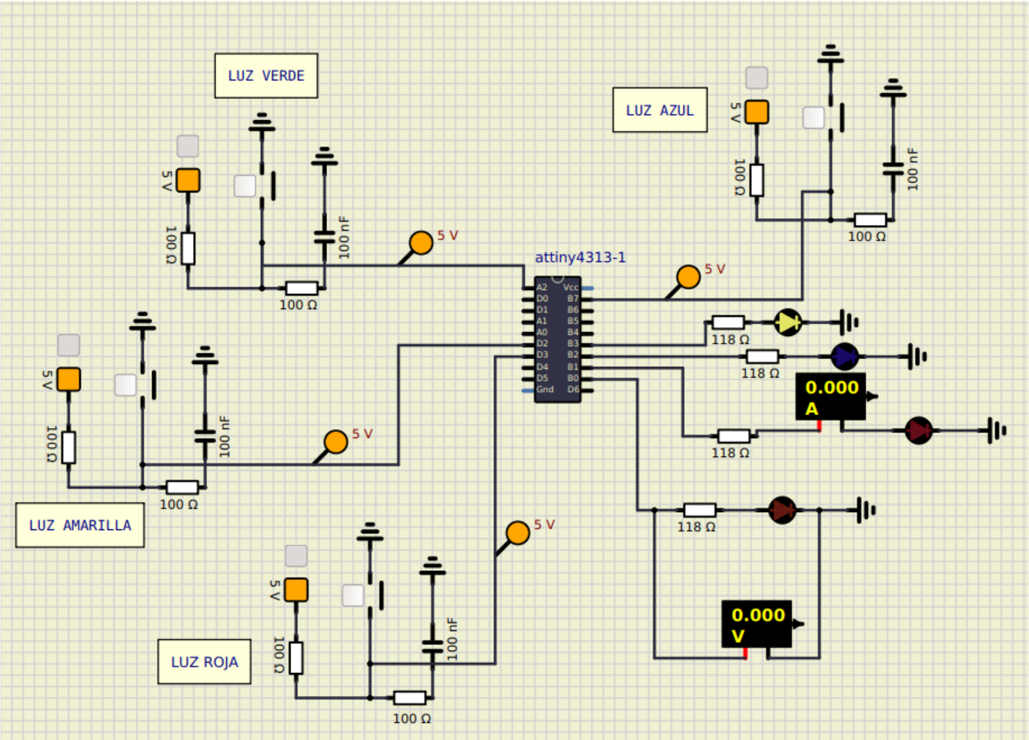


Figura 17: Funcionamiento de LED amarillo.

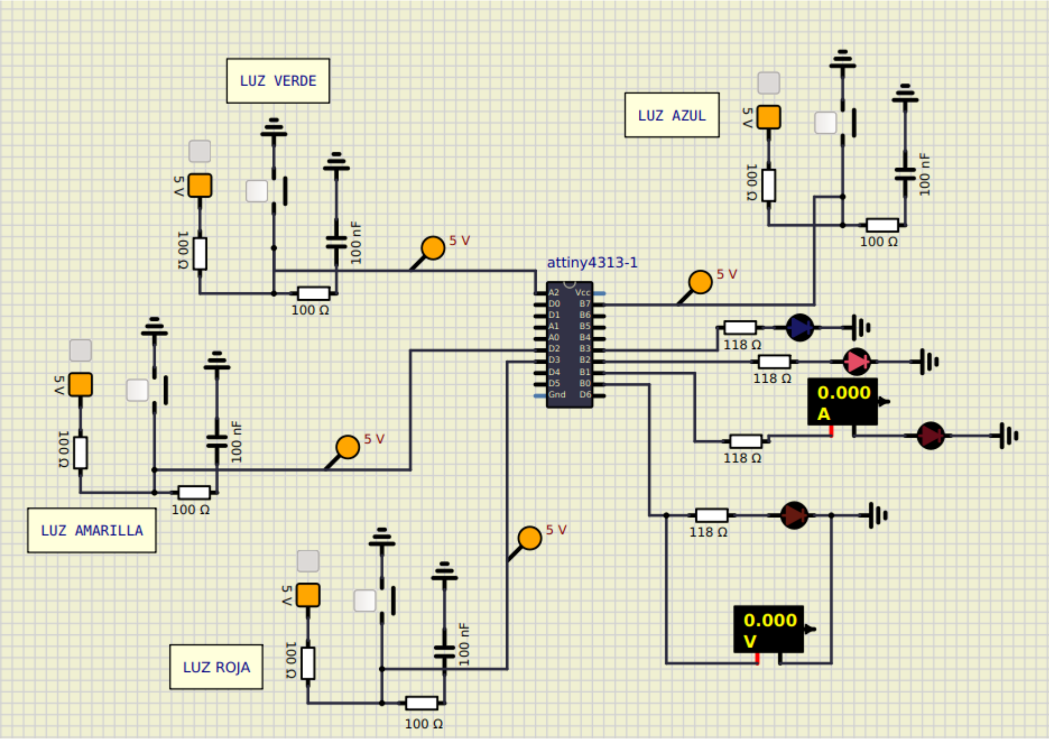


Figura 18: Funcionamiento de LED rojo.

3.4. Temporización de los LEDs

Para poder mantener la cuenta del tiempo que los LEDs permanecen encendidos se crea una función llamada *delay()*, la cual recibe como parámetro la cantidad de **interrupts** que queremos que pasen, y se muestra en el código 3.

El diseño o raciocinio detrás de esta función fue el siguiente: Se sabe que el microcontrolador ATTiny-4313 opera a un reloj de 8 Mhz. A esa velocidad por defecto se le aplicó un prescaling de 64, de forma que la nueva frecuencia del reloj es de 0.125 Mhz y, por lo tanto, el período corresponde a $8\mu s$. Con estos datos, se sabe que para que un overflow ocurra, el contador TCNT0 debe de contar 256 veces (de 0 a 255 ya que es de 8 bits), por lo que deben de pasar $256 \cdot 8\mu s = 2{,}048ms$ para ello. Con esto en mente, podemos calcular fácilmente cuántos overflows deben ocurrir para obtener un cierto delay. En este laboratorio, los LEDs deben de permanecer encendidos 2 segundos al principio y con cada avance en la dificultad se debe ir restando 200 ms, para ello hacemos un cálculo rápido:

$$delay_{2s} = \frac{2}{2{,}048 \times 10^{-3}} = 976{,}56 \approx 977 \tag{5}$$

y para ir restando los 200 ms:

$$delay_{200ms} = \frac{2 \times 10^{-3}}{2{,}048 \times 10^{-3}} = 97{,}66 \approx 98 \tag{6}$$

Por lo tanto, deben de ocurrir 977 overflows para que pasen 2 segundos y debemos restarle estos unos 98 overflows cada vez que se sube de nivel para ir restando 200 ms. Al comenzar el juego y al ir avanzando en los niveles, se podrá notar que los LEDs duran encendidos cada vez menos tiempo, poniendo en evidencia que la implementación realizada funciona sin problemas.

```
void delay(int overflows){
  TIMSK = 0b10;
  enable = 1;
  units_counter = 0;
  TCNT0 = 0x00;
  while(units_counter < overflows){
    PORTB &= 0xFF; // No hace nada pero un while vacio genera problemas
  }
  TIMSK = 0b00;
  enable = 0;
}
...

ISR(TIMER0_OVF_vect)
{
  if (enable) units_counter++;
}
...
```

Código 3: Función `delay(int overflows)`

4. Conclusiones y recomendaciones

- Se recomienda leer cuidadosamente la hoja del fabricante para el uso correcto de los pines.
- Se sugiere siempre tener los elementos conectados a tierra pues a pesar de ser una simulación siempre es importante tener en cuenta el cuidado de los componentes electrónicos.
- Dentro del análisis expuesto, es posible afirmar el buen funcionamiento de las interrupciones del programa.

Referencias

[1] Microchip Technology Inc. Microchip technology.

[2] Isaac. Pulsador: Cómo usar este simple elemento con arduino, Oct 2019.

[3] Steren. Venta de potenciómetros y resistencias.

5. Anexos

5.1. Repositorio Git

En este repositorio `eusanchez y amonbon GIT`, se adjunta el proyecto donde se puede observar el progreso y los archivos de `.simu` y `.c`.

5.2. Hojas del Fabricante del ATtiny2313

Features

- High Performance, Low Power AVR® 8-Bit Microcontroller
- Advanced RISC Architecture
  - 120 Powerful Instructions – Most Single Clock Cycle Execution
  - 32 x 8 General Purpose Working Registers
  - Fully Static Operation
  - Up to 20 MIPS Throughput at 20 MHz
- Data and Non-volatile Program and Data Memories
  - 2/4K Bytes of In-System Self Programmable Flash
    - Endurance 10,000 Write/Erase Cycles
  - 128/256 Bytes In-System Programmable EEPROM
    - Endurance: 100,000 Write/Erase Cycles
  - 128/256 Bytes Internal SRAM
  - Programming Lock for Flash Program and EEPROM Data Security
- Peripheral Features
  - One 8-bit Timer/Counter with Separate Prescaler and Compare Mode
  - One 16-bit Timer/Counter with Separate Prescaler, Compare and Capture Modes
  - Four PWM Channels
  - On-chip Analog Comparator
  - Programmable Watchdog Timer with On-chip Oscillator
  - USI – Universal Serial Interface
  - Full Duplex USART
- Special Microcontroller Features
  - debugWIRE On-chip Debugging
  - In-System Programmable via SPI Port
  - External and Internal Interrupt Sources
  - Low-power Idle, Power-down, and Standby Modes
  - Enhanced Power-on Reset Circuit
  - Programmable Brown-out Detection Circuit
  - Internal Calibrated Oscillator
- I/O and Packages
  - 18 Programmable I/O Lines
  - 20-pin PDIP, 20-pin SOIC, 20-pad MLF/VQFN
- Operating Voltage
  - 1.8 – 5.5V
- Speed Grades
  - 0 – 4 MHz @ 1.8 – 5.5V
  - 0 – 10 MHz @ 2.7 – 5.5V
  - 0 – 20 MHz @ 4.5 – 5.5V
- Industrial Temperature Range: -40°C to +85°C
- Low Power Consumption
  - Active Mode
    - 190 µA at 1.8V and 1MHz
  - Idle Mode
    - 24 µA at 1.8V and 1MHz
  - Power-down Mode
    - 0.1 µA at 1.8V and +25°C

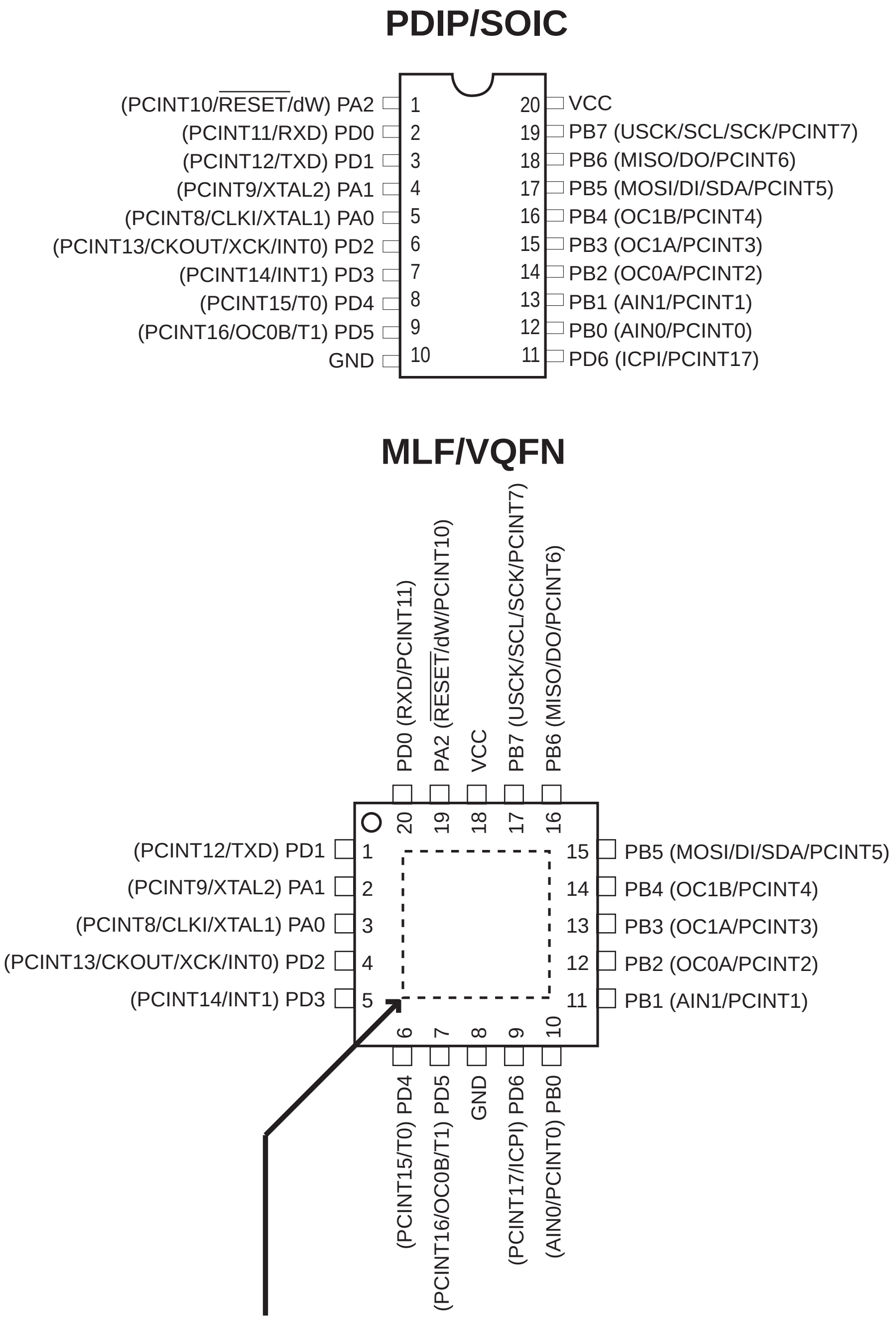
8-bit AVR® Microcontroller with 2/4K Bytes In-System Programmable Flash

ATtiny2313A ATtiny4313

Rev. 8246B–AVR–09/11

1. Pin Configurations

Figure 1-1. Pinout ATtiny2313A/4313



NOTE: Bottom pad should be soldered to ground.

1.1 Pin Descriptions

1.1.1 VCC

Digital supply voltage.

1.1.2 GND

Ground.

1.1.3 Port A (PA2..PA0)

Port A is a 3-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port A output buffers have symmetrical drive characteristics with both high sink and source capability, except PA2 which has the $\overline{\text{RESET}}$ capability. To use pin PA2 as I/O pin, instead of RESET pin, program (“0”) RSTDISBL fuse. As inputs, Port A pins that are externally pulled low will source current if the pull-up resistors are activated. The Port A pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Port A also serves the functions of various special features of the ATtiny2313A/4313 as listed on page 62.

1.1.4 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 also serves the functions of various special features of the ATtiny2313A/4313 as listed on page 63.

1.1.5 Port D (PD6..PD0)

Port D is a 7-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 ATtiny2313A/4313 as listed on page 67.

1.1.6 $\overline{\text{RESET}}$

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 and provided that the reset pin has not been disabled. The minimum pulse length is given in Table 22-3 on page 201. Shorter pulses are not guaranteed to generate a reset. The Reset Input is an alternate function for PA2 and dW.

The reset pin can also be used as a (weak) I/O pin.

1.1.7 XTAL1

Input to the inverting Oscillator amplifier and input to the internal clock operating circuit. XTAL1 is an alternate function for PA0.

1.1.8 XTAL2

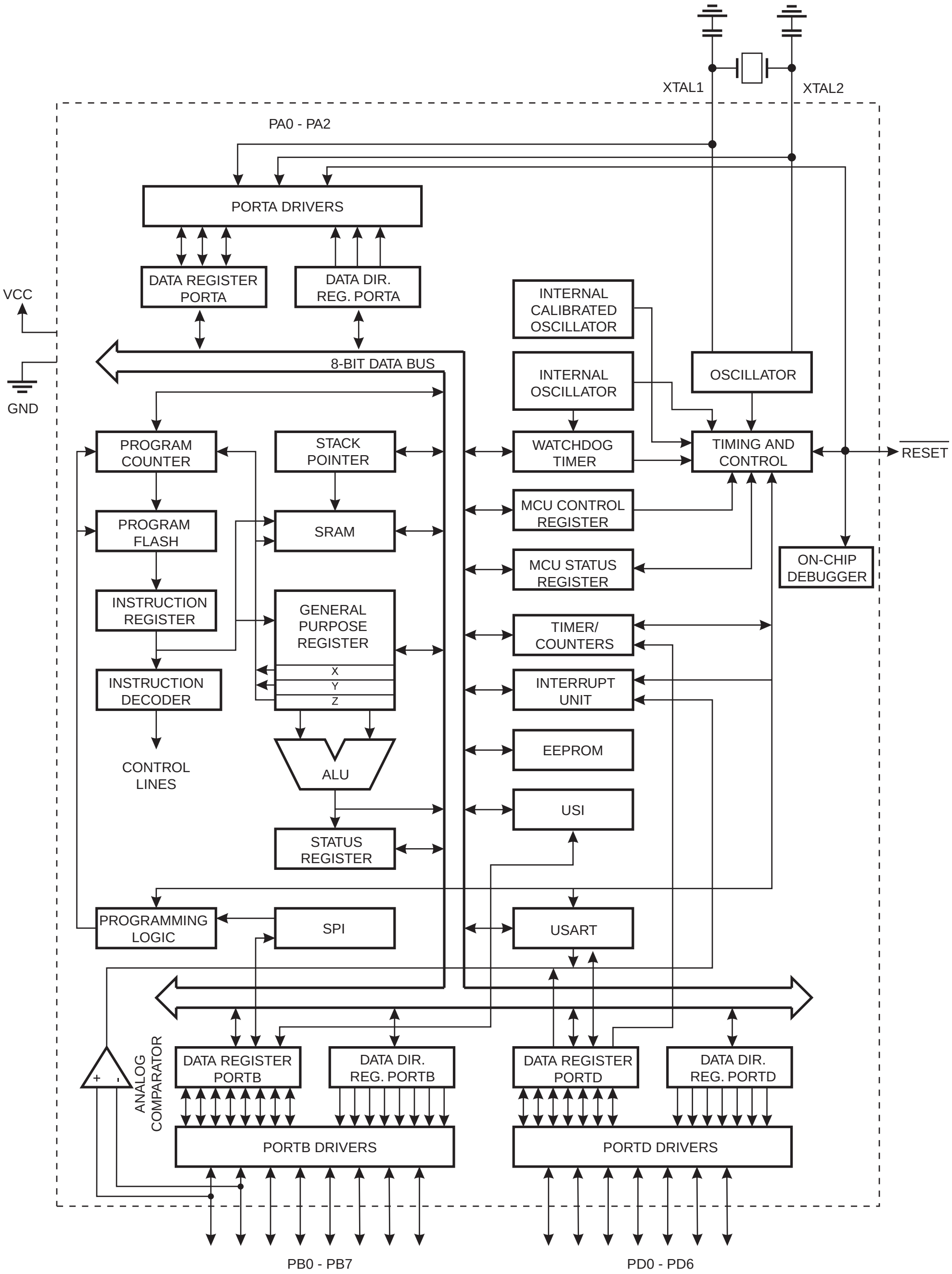
Output from the inverting Oscillator amplifier. XTAL2 is an alternate function for PA1.

2. Overview

The ATtiny2313A/4313 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 ATtiny2313A/4313 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 ATtiny2313A/4313 provides the following features: 2/4K bytes of In-System Programmable Flash, 128/256 bytes EEPROM, 128/256 bytes SRAM, 18 general purpose I/O lines, 32 general purpose working registers, a single-wire Interface for On-chip Debugging, two flexible Timer/Counters with compare modes, internal and external interrupts, a serial programmable USART, Universal Serial Interface with Start Condition Detector, a programmable Watchdog Timer with internal Oscillator, and three software selectable power saving modes. The Idle mode stops the CPU while allowing the SRAM, Timer/Counters, 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 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 non-volatile memory technology. The On-chip ISP Flash allows the program memory to be reprogrammed In-System through an SPI serial interface, or by a conventional non-volatile memory programmer. By combining an 8-bit RISC CPU with In-System Self-Programmable Flash on a monolithic chip, the Atmel ATtiny2313A/4313 is a powerful microcontroller that provides a highly flexible and cost effective solution to many embedded control applications.

The ATtiny2313A/4313 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 ATtiny2313A and ATtiny4313

The ATtiny2313A and ATtiny4313 differ only in memory sizes. Table 2-1 summarizes the different memory sizes for the two devices.

Table 2-1. Memory Size Summary

Device	Flash	EEPROM	RAM
ATtiny2313A	2K Bytes	128 Bytes	128 Bytes
ATtiny4313	4K Bytes	256 Bytes	256 Bytes

3. About

3.1 Resources

A comprehensive set of drivers, application notes, data sheets and descriptions on development tools are available for download at http://www.atmel.com/avr.

3.2 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 the 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, this means “LDS” and “STS” combined with “SBRS”, “SBRC”, “SBR”, and “CBR”. Note that not all AVR devices include an extended I/O map.

3.3 Data Retention

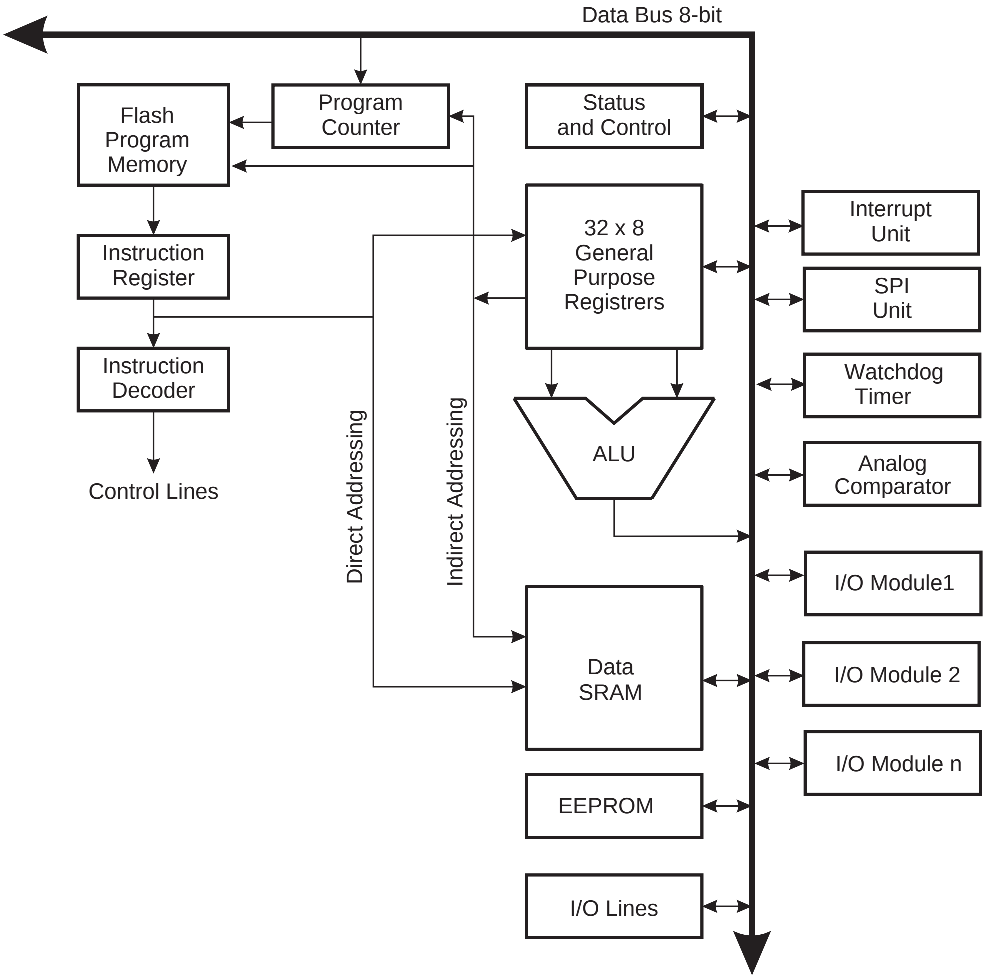
Reliability Qualification results show that the projected data retention failure rate is much less than 1 PPM over 20 years at 85°C or 100 years at 25°C.

4. CPU Core

This section discusses the AVR core architecture in general. The main function of the CPU core is to ensure correct program execution. The CPU must therefore be able to access memories, perform calculations, control peripherals, and handle interrupts.

4.1 Architectural Overview

Figure 4-1. Block Diagram of the AVR Architecture



In order to maximize performance and parallelism, the AVR uses a Harvard architecture – with separate memories and buses for program and data. Instructions in the program memory are executed with a single level pipelining. While one instruction is being executed, the next instruction is pre-fetched from the program memory. This concept enables instructions to be executed in every clock cycle. The program memory is In-System Reprogrammable Flash memory.

The fast-access Register File contains 32 x 8-bit general purpose working registers with a single clock cycle access time. This allows single-cycle Arithmetic Logic Unit (ALU) operation. In a typical ALU operation, two operands are output from the Register File, the operation is executed, and the result is stored back in the Register File – in one clock cycle.

Six of the 32 registers can be used as three 16-bit indirect address register pointers for Data Space addressing – enabling efficient address calculations. One of the these address pointers can also be used as an address pointer for look up tables in Flash program memory. These added function registers are the 16-bit X-, Y-, and Z-register, described later in this section.

The ALU supports arithmetic and logic operations between registers or between a constant and a register. Single register operations can also be executed in the ALU. After an arithmetic operation, the Status Register is updated to reflect information about the result of the operation.

Program flow is provided by conditional and unconditional jump and call instructions, able to directly address the whole address space. Most AVR instructions have a single 16-bit word format. Every program memory address contains a 16- or 32-bit instruction.

During interrupts and subroutine calls, the return address Program Counter (PC) is stored on the Stack. The Stack is effectively allocated in the general data SRAM, and consequently the Stack size is only limited by the total SRAM size and the usage of the SRAM. All user programs must initialize the SP in the Reset routine (before subroutines or interrupts are executed). The Stack Pointer (SP) is read/write accessible in the I/O space. The data SRAM can easily be accessed through the five different addressing modes supported in the AVR architecture.

The memory spaces in the AVR architecture are all linear and regular memory maps.

A flexible interrupt module has its control registers in the I/O space with an additional Global Interrupt Enable bit in the Status Register. All interrupts have a separate Interrupt Vector in the Interrupt Vector table. The interrupts have priority in accordance with their Interrupt Vector position. The lower the Interrupt Vector address, the higher the priority.

The I/O memory space contains 64 addresses for CPU peripheral functions as Control Registers, and other I/O functions. The I/O Memory can be accessed directly, or as the Data Space locations following those of the Register File, 0x20 - 0x5F.

4.2 ALU – Arithmetic Logic Unit

The high-performance AVR ALU operates in direct connection with all the 32 general purpose working registers. Within a single clock cycle, arithmetic operations between general purpose registers or between a register and an immediate are executed. The ALU operations are divided into three main categories – arithmetic, logical, and bit-functions. Some implementations of the architecture also provide a powerful multiplier supporting both signed/unsigned multiplication and fractional format. See the “Instruction Set” section for a detailed description.

4.3 Status Register

The Status Register contains information about the result of the most recently executed arithmetic instruction. This information can be used for altering program flow in order to perform conditional operations. Note that the Status Register is updated after all ALU operations, as specified in the Instruction Set Reference. This will in many cases remove the need for using the dedicated compare instructions, resulting in faster and more compact code.

The Status Register is not automatically stored when entering an interrupt routine and restored when returning from an interrupt. This must be handled by software.

The AVR Status Register – SREG – is defined as:

Bit	7	6	5	4	3	2	1	0	
0x3F (0x5F)	I	T	H	S	V	N	Z	C	SREG
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7 – I: Global Interrupt Enable**

The Global Interrupt Enable bit must be set for the interrupts to be enabled. The individual interrupt enable control is then performed in separate control registers. If the Global Interrupt Enable Register is cleared, none of the interrupts are enabled independent of the individual interrupt enable settings. The I-bit is cleared by hardware after an interrupt has occurred, and is set by the RETI instruction to enable subsequent interrupts. The I-bit can also be set and cleared by the application with the SEI and CLI instructions, as described in the instruction set reference.

- **Bit 6 – T: Bit Copy Storage**

The Bit Copy instructions BLD (Bit LoaD) and BST (Bit STore) use the T-bit as source or destination for the operated bit. A bit from a register in the Register File can be copied into T by the BST instruction, and a bit in T can be copied into a bit in a register in the Register File by the BLD instruction.

- **Bit 5 – H: Half Carry Flag**

The Half Carry Flag H indicates a Half Carry in some arithmetic operations. Half Carry Is useful in BCD arithmetic. See the “Instruction Set Description” for detailed information.

- **Bit 4 – S: Sign Bit, $S = N \oplus V$**

The S-bit is always an exclusive or between the negative flag N and the Two’s Complement Overflow Flag V. See the “Instruction Set Description” for detailed information.

- **Bit 3 – V: Two’s Complement Overflow Flag**

The Two’s Complement Overflow Flag V supports two’s complement arithmetics. See the “Instruction Set Description” for detailed information.

- **Bit 2 – N: Negative Flag**

The Negative Flag N indicates a negative result in an arithmetic or logic operation. See the “Instruction Set Description” for detailed information.

- **Bit 1 – Z: Zero Flag**

The Zero Flag Z indicates a zero result in an arithmetic or logic operation. See the “Instruction Set Description” for detailed information.

- **Bit 0 – C: Carry Flag**

The Carry Flag C indicates a carry in an arithmetic or logic operation. See the “Instruction Set Description” for detailed information.

9. Interrupts

This section describes the specifics of the interrupt handling as performed in ATtiny2313A/4313. For a general explanation of the AVR interrupt handling, refer to "Reset and Interrupt Handling" on page 13.

9.1 Interrupt Vectors

The interrupt vectors of ATtiny2313A/4313 are described in Table 9-1 below

Table 9-1. Reset and Interrupt Vectors

Vector No.	Program Address	Label	Interrupt Source
1	0x0000	RESET	External Pin, Power-on Reset, Brown-out Reset, and Watchdog Reset
2	0x0001	INT0	External Interrupt Request 0
3	0x0002	INT1	External Interrupt Request 1
4	0x0003	TIMER1 CAPT	Timer/Counter1 Capture Event
5	0x0004	TIMER1 COMPA	Timer/Counter1 Compare Match A
6	0x0005	TIMER1 OVF	Timer/Counter1 Overflow
7	0x0006	TIMER0 OVF	Timer/Counter0 Overflow
8	0x0007	USART0, RX	USART0, Rx Complete
9	0x0008	USART0, UDRE	USART0 Data Register Empty
10	0x0009	USART0, TX	USART0, Tx Complete
11	0x000A	ANALOG COMP	Analog Comparator
12	0x000B	PCINT0	Pin Change Interrupt Request 0
13	0x000C	TIMER1 COMPB	Timer/Counter1 Compare Match B
14	0x000D	TIMER0 COMPA	Timer/Counter0 Compare Match A
15	0x000E	TIMER0 COMPB	Timer/Counter0 Compare Match B
16	0x000F	USI START	USI Start Condition
17	0x0010	USI OVERFLOW	USI Overflow
18	0x0011	EE READY	EEPROM Ready
19	0x0012	WDT OVERFLOW	Watchdog Timer Overflow
20	0x0013	PCINT1	Pin Change Interrupt Request 1
21	0x0014	PCINT2	Pin Change Interrupt Request 2

In case the program never enables an interrupt source, the Interrupt Vectors will not be used and, consequently, regular program code can be placed at these locations.

The most typical and general setup for the Interrupt Vector Addresses in ATtiny2313A/4313 shown below:

```
Address  Labels Code                Comments
0x0000          rjmp   RESET          ; Reset Handler
0x0001          rjmp   INT0           ; External Interrupt0 Handler
0x0002          rjmp   INT1           ; External Interrupt1 Handler
0x0003          rjmp   TIM1_CAPT      ; Timer1 Capture Handler
0x0004          rjmp   TIM1_COMPA     ; Timer1 CompareA Handler
0x0005          rjmp   TIM1_OVF       ; Timer1 Overflow Handler
0x0006          rjmp   TIM0_OVF       ; Timer0 Overflow Handler
0x0007          rjmp   USART0_RXC     ; USART0 RX Complete Handler
0x0008          rjmp   USART0_DRE     ; USART0,UDR Empty Handler
0x0009          rjmp   USART0_TXC     ; USART0 TX Complete Handler
0x000A          rjmp   ANA_COMP       ; Analog Comparator Handler
0x000B          rjmp   PCINT0         ; PCINT0 Handler
0x000C          rjmp   TIMER1_COMPB   ; Timer1 Compare B Handler
0x000D          rjmp   TIMER0_COMPA   ; Timer0 Compare A Handler
0x000E          rjmp   TIMER0_COMPB   ; Timer0 Compare B Handler
0x000F          rjmp   USI_START      ; USI Start Handler
0x0010          rjmp   USI_OVERFLOW   ; USI Overflow Handler
0x0011          rjmp   EE_READY       ; EEPROM Ready Handler
0x0012          rjmp   WDT_OVERFLOW   ; Watchdog Overflow Handler
0x0013          rjmp   PCINT1         ; PCINT1 Handler
0x0014          rjmp   PCINT2         ; PCINT2 Handler
;
0x0013   RESET: ldi    r16, low(RAMEND); Main program start
0x0014          out    SPL,r16          Set Stack Pointer to top of RAM
0x0015          sei                   ; Enable interrupts
0x0016          <instr>  xxx
  ...    ...    ...  ...
```

9.2 External Interrupts

External Interrupts are triggered by the INT0 or INT1 pin or any of the PCINT17..0 pins. Observe that, if enabled, the interrupts will trigger even if the INT0, INT1 or PCINT17..0 pins are configured as outputs. This feature provides a way of generating a software interrupt. Pin change 0 interrupts PCI0 will trigger if any enabled PCINT7..0 pin toggles. Pin change 1 interrupts PCI1 will trigger if any enabled PCINT10..8 pin toggles. Pin change 2 interrupts PCI2 will trigger, if any enabled PCINT17..11 pin toggles. The PCMSK0, PCMSK1, and PCMSK2 Registers control which pins contribute to the pin change interrupts. Pin change interrupts on PCINT17..0 are detected asynchronously, which means that these interrupts can be used for waking the part also from sleep modes other than Idle mode.

The INT0 and INT1 interrupts can be triggered by a falling or rising edge or a low level. This is set up as shown in "MCUCR – MCU Control Register" on page 51. When the INT0 or INT1 interrupt is enabled and configured as level triggered, the interrupt will trigger as long as the pin is held low. Note that recognition of falling or rising edge interrupts on INT0 or INT1 requires the presence of an I/O clock, as described in "Clock Sources" on page 27.

9.2.1 Low Level Interrupt

A low level interrupt on INT0 or INT1 is detected asynchronously. This means that the interrupt source can be used for waking the part also from sleep modes other than Idle (the I/O clock is halted in all sleep modes except Idle).

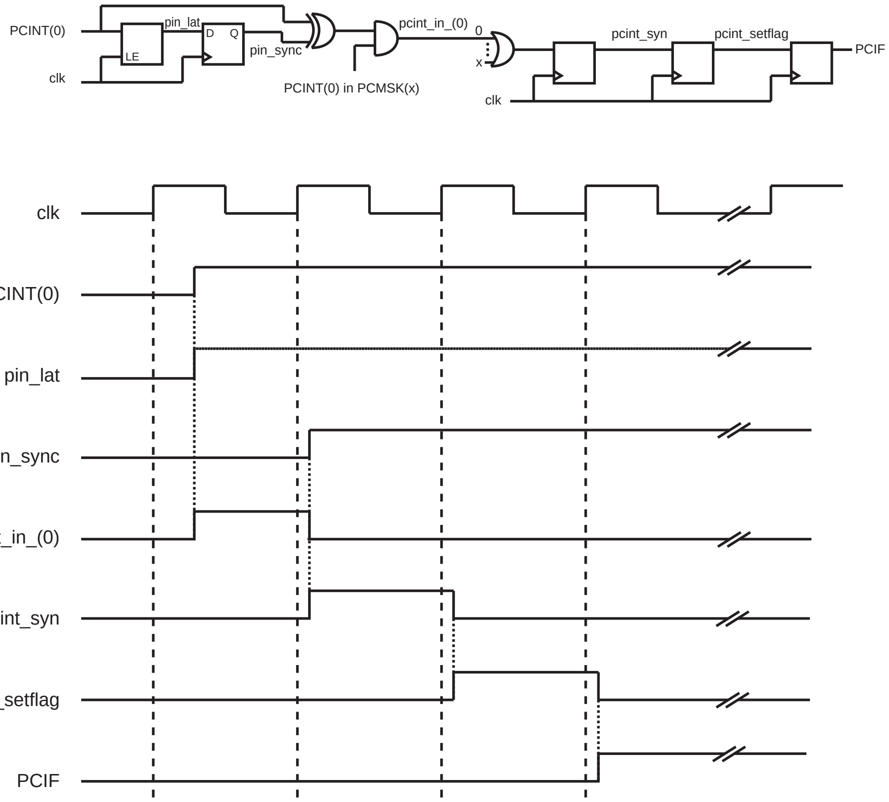
Note that if a level triggered interrupt is used for wake-up from Power-down, the required level must be held long enough for the MCU to complete the wake-up to trigger the level interrupt. If the level disappears before the end of the Start-up Time, the MCU will still wake up, but no interrupt will be generated. The start-up time is defined by the SUT and CKSEL fuses, as described in "Clock System" on page 26.

If the low level on the interrupt pin is removed before the device has woken up then program execution will not be diverted to the interrupt service routine but continue from the instruction following the SLEEP command.

9.2.2 Pin Change Interrupt Timing

A timing example of a pin change interrupt is shown in Figure 9-1.

Figure 9-1. Timing of pin change interrupts



9.3 Register Description

9.3.1 MCUCR – MCU Control Register

The External Interrupt Control Register contains control bits for interrupt sense control.

Bit	7	6	5	4	3	2	1	0	
0x35 (0x55)	PUD	SM1	SE	SM0	ISC11	ISC10	ISC01	ISC00	MCUCR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 3, 2 – ISC11, ISC10: Interrupt Sense Control 1 Bit 1 and Bit 0**

The External Interrupt 1 is activated by the external pin INT1 if the SREG I-flag and the corresponding interrupt mask are set. The level and edges on the external INT1 pin that activate the interrupt are defined in Table 9-2. The value on the INT1 pin is sampled before detecting edges. If edge or toggle interrupt is selected, pulses that last longer than one clock period will generate an interrupt. Shorter pulses are not guaranteed to generate an interrupt. If low level interrupt is selected, the low level must be held until the completion of the currently executing instruction to generate an interrupt

Table 9-2. Interrupt 1 Sense Control

ISC11	ISC10	Description
0	0	The low level of INT1 generates an interrupt request.
0	1	Any logical change on INT1 generates an interrupt request.
1	0	The falling edge of INT1 generates an interrupt request.
1	1	The rising edge of INT1 generates an interrupt request.

- **Bits 1, 0 – ISC01, ISC00: Interrupt Sense Control 0 Bit 1 and Bit 0**

The External Interrupt 0 is activated by the external pin INT0 if the SREG I-flag and the corresponding interrupt mask are set. The level and edges on the external INT0 pin that activate the interrupt are defined in Table 9-3. The value on the INT0 pin is sampled before detecting edges. If edge or toggle interrupt is selected, pulses that last longer than one clock period will generate an interrupt. Shorter pulses are not guaranteed to generate an interrupt. If low level interrupt is selected, the low level must be held until the completion of the currently executing instruction to generate an interrupt.

Table 9-3. Interrupt 0 Sense Control

ISC01	ISC00	Description
0	0	The low level of INT0 generates an interrupt request.
0	1	Any logical change on INT0 generates an interrupt request.
1	0	The falling edge of INT0 generates an interrupt request.
1	1	The rising edge of INT0 generates an interrupt request.

9.3.2 GIMSK – General Interrupt Mask Register

Bit	7	6	5	4	3	2	1	0	
0x3B (0x5B)	**INT1**	**INT0**	**PCIE0**	**PCIE2**	**PCIE1**	**–**	**–**	**–**	**GIMSK**
Read/Write	R/W	R/W	R/W	R/W	R/W	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	

- **Bits 2..0 – Res: Reserved Bits**

These bits are reserved and will always read as zero.

- **Bit 7 – INT1: External Interrupt Request 1 Enable**

When the INT1 bit is set (one) and the I-bit in the Status Register (SREG) is set (one), the external pin interrupt is enabled. The Interrupt Sense Control bits (ISC11 and ISC10) in the External Interrupt Control Register A (EICRA) define whether the external interrupt is activated on rising and/or falling edge of the INT1 pin or level sensed. Activity on the pin will cause an interrupt request even if INT1 is configured as an output. The corresponding interrupt of External Interrupt Request 1 is executed from the INT1 Interrupt Vector.

- **Bit 6 – INT0: External Interrupt Request 0 Enable**

When the INT0 bit is set (one) and the I-bit in the Status Register (SREG) is set (one), the external pin interrupt is enabled. The Interrupt Sense Control bits (ISC01 and ISC00) in the External Interrupt Control Register A (EICRA) define whether the external interrupt is activated on rising and/or falling edge of the INT0 pin or level sensed. Activity on the pin will cause an interrupt request even if INT0 is configured as an output. The corresponding interrupt of External Interrupt Request 0 is executed from the INT0 Interrupt Vector.

- **Bit 5 – PCIE0: Pin Change Interrupt Enable 0**

When the PCIE0 bit is set (one) and the I-bit in the Status Register (SREG) is set (one), pin change interrupt 0 is enabled. Any change on any enabled PCINT7..0 pin will cause an interrupt. The corresponding interrupt of Pin Change Interrupt Request is executed from the PCI0 Interrupt Vector. PCINT7..0 pins are enabled individually by the PCMSK0 Register.

- **Bit 4 – PCIE2: Pin Change Interrupt Enable 2**

When the PCIE2 bit is set (one) and the I-bit in the Status Register (SREG) is set (one), pin change interrupt 2 is enabled. Any change on any enabled PCINT17..11 pin will cause an interrupt. The corresponding interrupt of Pin Change Interrupt Request is executed from the PCI2 Interrupt Vector. PCINT17..11 pins are enabled individually by the PCMSK2 Register.

- **Bit 3 – PCIE1: Pin Change Interrupt Enable 1**

When the PCIE1 bit is set (one) and the I-bit in the Status Register (SREG) is set (one), pin change interrupt 1 is enabled. Any change on any enabled PCINT10..8 pin will cause an interrupt. The corresponding interrupt of Pin Change Interrupt Request is executed from the PCI1 Interrupt Vector. PCINT10..8 pins are enabled individually by the PCMSK1 Register.

9.3.3 GIFR – General Interrupt Flag Register

Bit	7	6	5	4	3	2	1	0	
0x3A (0x5A)	INTF1	INTF0	PCIF0	PCIF2	PCIF1	–	–	–	GIFR
Read/Write	R/W	R/W	R/W	R/W	R/W	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	

- **Bits 2..0 – Res: Reserved Bits**

These bits are reserved and will always read as zero.

- **Bit 7 – INTF1: External Interrupt Flag 1**

When an edge or logic change on the INT1 pin triggers an interrupt request, INTF1 becomes set (one). If the I-bit in SREG and the INT1 bit in GIMSK are set (one), the MCU will jump to the corresponding Interrupt Vector. The flag is cleared when the interrupt routine is executed. Alternatively, the flag can be cleared by writing a logical one to it. This flag is always cleared when INT1 is configured as a level interrupt.

- **Bit 6 – INTF0: External Interrupt Flag 0**

When an edge or logic change on the INT0 pin triggers an interrupt request, INTF0 becomes set (one). If the I-bit in SREG and the INT0 bit in GIMSK are set (one), the MCU will jump to the corresponding Interrupt Vector. The flag is cleared when the interrupt routine is executed. Alternatively, the flag can be cleared by writing a logical one to it. This flag is always cleared when INT0 is configured as a level interrupt.

- **Bit 5 – PCIF0: Pin Change Interrupt Flag 0**

When a logic change on any PCINT7..0 pin triggers an interrupt request, PCIF becomes set (one). If the I-bit in SREG and the PCIE0 bit in GIMSK are set (one), the MCU will jump to the corresponding Interrupt Vector. The flag is cleared when the interrupt routine is executed. Alternatively, the flag can be cleared by writing a logical one to it.

- **Bit 4 – PCIF2: Pin Change Interrupt Flag 2**

When a logic change on any PCINT17..11 pin triggers an interrupt request, PCIF2 becomes set (one). If the I-bit in SREG and the PCIE2 bit in GIMSK are set (one), the MCU will jump to the corresponding Interrupt Vector. The flag is cleared when the interrupt routine is executed. Alternatively, the flag can be cleared by writing a logical one to it.

- **Bit 3 – PCIF1: Pin Change Interrupt Flag 1**

When a logic change on any PCINT10..8 pin triggers an interrupt request, PCIF1 becomes set (one). If the I-bit in SREG and the PCIE1 bit in GIMSK are set (one), the MCU will jump to the corresponding Interrupt Vector. The flag is cleared when the interrupt routine is executed. Alternatively, the flag can be cleared by writing a logical one to it.

9.3.4 PCMSK2 – Pin Change Mask Register 2

Bit	7	6	5	4	3	2	1	0	
0x05 (0x25)	–	PCINT17	PCINT16	PCINT15	PCINT14	PCINT13	PCINT12	PCINT11	PCMSK2
Read/Write	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7 – Res: Reserved Bit**

These bits are reserved and will always read as zero.

- **Bits 6..0 – PCINT17..11: Pin Change Enable Mask 17..11**

Each PCINT17..11 bit selects whether pin change interrupt is enabled on the corresponding I/O pin. If PCINT17..11 is set and the PCIE1 bit in GIMSK is set, pin change interrupt is enabled on the corresponding I/O pin. If PCINT17..11 is cleared, pin change interrupt on the corresponding I/O pin is disabled.

9.3.5 PCMSK1 – Pin Change Mask Register 1

Bit	7	6	5	4	3	2	1	0	
0x04 (0x24)	–	–	–	–	–	PCINT10	PCINT9	PCINT8	PCMSK1
Read/Write	R	R	R	R	R	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bits 7:3 – Res: Reserved Bits**

These bits are reserved and will always read as zero.

- **Bits 2..0 – PCINT10..8: Pin Change Enable Mask 10..8**

Each PCINT10..8 bit selects whether pin change interrupt is enabled on the corresponding I/O pin. If PCINT10..8 is set and the PCIE1 bit in GIMSK is set, pin change interrupt is enabled on the corresponding I/O pin. If PCINT10..8 is cleared, pin change interrupt on the corresponding I/O pin is disabled.

9.3.6 PCMSK0 – Pin Change Mask Register 0

Bit	7	6	5	4	3	2	1	0	
0x20 (0x40)	PCINT7	PCINT6	PCINT5	PCINT4	PCINT3	PCINT2	PCINT1	PCINT0	PCMSK0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

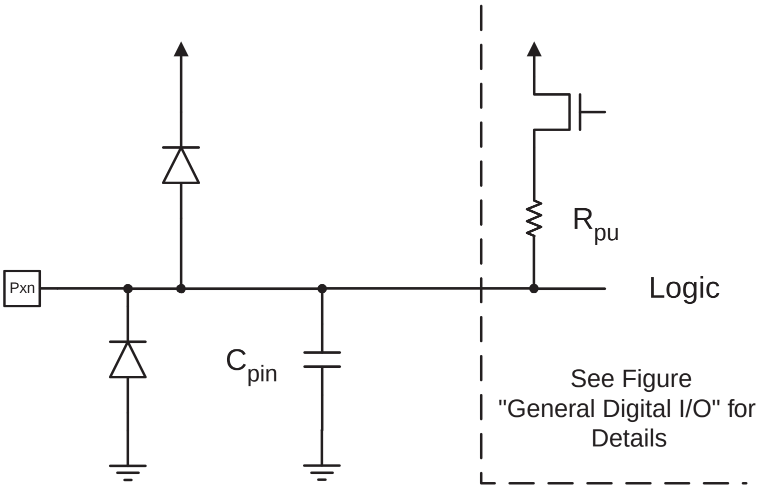
- **Bits 7..0 – PCINT7..0: Pin Change Enable Mask 7..0**

Each PCINT7..0 bit selects whether pin change interrupt is enabled on the corresponding I/O pin. If PCINT7..0 is set and the PCIE0 bit in GIMSK is set, pin change interrupt is enabled on the corresponding I/O pin. If PCINT7..0 is cleared, pin change interrupt on the corresponding I/O pin is disabled.

10. I/O-Ports

All AVR ports have true Read-Modify-Write functionality when used as general digital I/O ports. This means that the direction of one port pin can be changed without unintentionally changing the direction of any other pin with the SBI and CBI instructions. The same applies when changing drive value (if configured as output) or enabling/disabling of pull-up resistors (if configured as input). Each output buffer has symmetrical drive characteristics with both high sink and source capability. The pin driver is strong enough to drive LED displays directly. All port pins have individually selectable pull-up resistors with a supply-voltage invariant resistance. All I/O pins have protection diodes to both V_{CC} and Ground as indicated in Figure 10-1 on page 55. See "Electrical Characteristics" on page 198 for a complete list of parameters.

Figure 10-1. I/O Pin Equivalent Schematic



All registers and bit references in this section are written in general form. A lower case "x" represents the numbering letter for the port, and a lower case "n" represents the bit number. However, when using the register or bit defines in a program, the precise form must be used. For example, PORTB3 for bit no. 3 in Port B, here documented generally as PORTxn. The physical I/O Registers and bit locations are listed in "Register Description" on page 69.

Three I/O memory address locations are allocated for each port, one each for the Data Register – PORTx, Data Direction Register – DDRx, and the Port Input Pins – PINx. The Port Input Pins I/O location is read only, while the Data Register and the Data Direction Register are read/write. However, writing a logic one to a bit in the PINx Register, will result in a toggle in the corresponding bit in the Data Register. In addition, the Pull-up Disable – PUD bit in MCUCR disables the pull-up function for all pins in all ports when set.

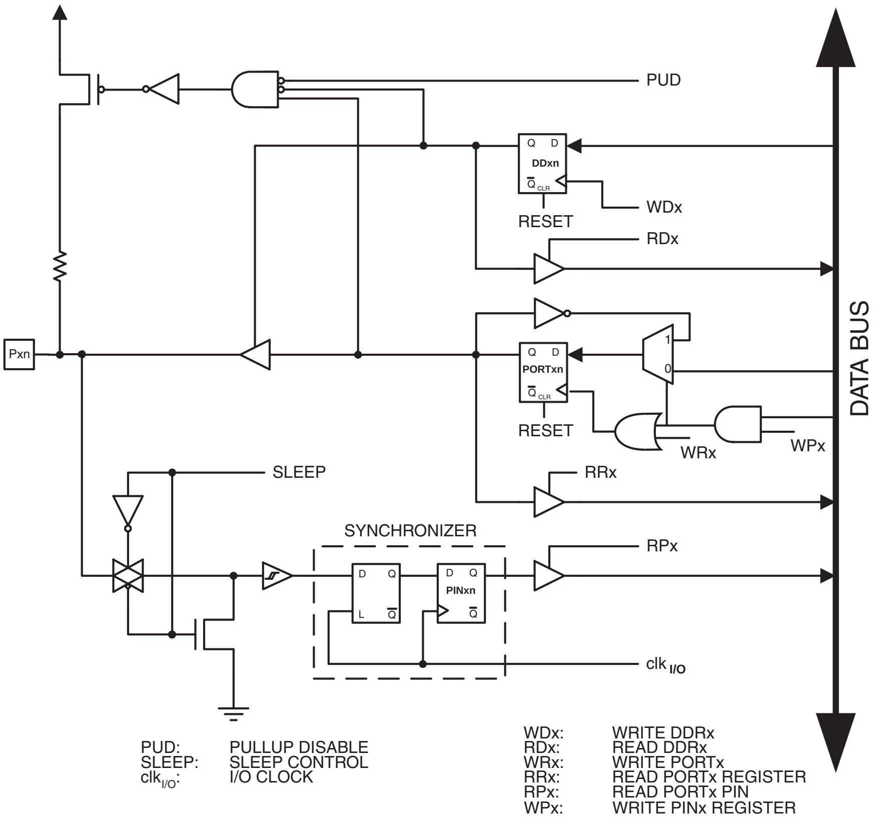
Using the I/O port as General Digital I/O is described in "Ports as General Digital I/O" on page 56. Most port pins are multiplexed with alternate functions for the peripheral features on the device. How each alternate function interferes with the port pin is described in "Alternate Port Functions" on page 60. Refer to the individual module sections for a full description of the alternate functions.

Note that enabling the alternate function of some of the port pins does not affect the use of the other pins in the port as general digital I/O.

10.1 Ports as General Digital I/O

The ports are bi-directional I/O ports with optional internal pull-ups. Figure 10-2 shows a functional description of one I/O-port pin, here generically called Pxn.

Figure 10-2. General Digital I/O[1]



Note: 1. WRx, WPx, WDx, RRx, RPx, and RDx are common to all pins within the same port. $clk_{I/O}$, SLEEP, and PUD are common to all ports.

10.1.1 Configuring the Pin

Each port pin consists of three register bits: DDxn, PORTxn, and PINxn. As shown in "Register Description" on page 69, the DDxn bits are accessed at the DDRx I/O address, the PORTxn bits at the PORTx I/O address, and the PINxn bits at the PINx I/O address.

The DDxn bit in the DDRx Register selects the direction of this pin. If DDxn is written logic one, Pxn is configured as an output pin. If DDxn is written logic zero, Pxn is configured as an input pin.

If PORTxn is written logic one when the pin is configured as an input pin, the pull-up resistor is activated. To switch the pull-up resistor off, PORTxn has to be written logic zero or the pin has to be configured as an output pin. The port pins are tri-stated when reset condition becomes active, even if no clocks are running.

If PORTxn is written logic one when the pin is configured as an output pin, the port pin is driven high (one). If PORTxn is written logic zero when the pin is configured as an output pin, the port pin is driven low (zero).

10.1.2 Toggling the Pin

Writing a logic one to PINxn toggles the value of PORTxn, independent on the value of DDRxn. Note that the SBI instruction can be used to toggle one single bit in a port.

10.1.3 Switching Between Input and Output

When switching between tri-state ({DDxn, PORTxn} = 0b00) and output high ({DDxn, PORTxn} = 0b11), an intermediate state with either pull-up enabled {DDxn, PORTxn} = 0b01) or output low ({DDxn, PORTxn} = 0b10) must occur. Normally, the pull-up enabled state is fully acceptable, as a high-impedant environment will not notice the difference between a strong high driver and a pull-up. If this is not the case, the PUD bit in the MCUCR Register can be set to disable all pull-ups in all ports.

Switching between input with pull-up and output low generates the same problem. The user must use either the tri-state ({DDxn, PORTxn} = 0b00) or the output high state ({DDxn, PORTxn} = 0b10) as an intermediate step.

Table 10-1 summarizes the control signals for the pin value.

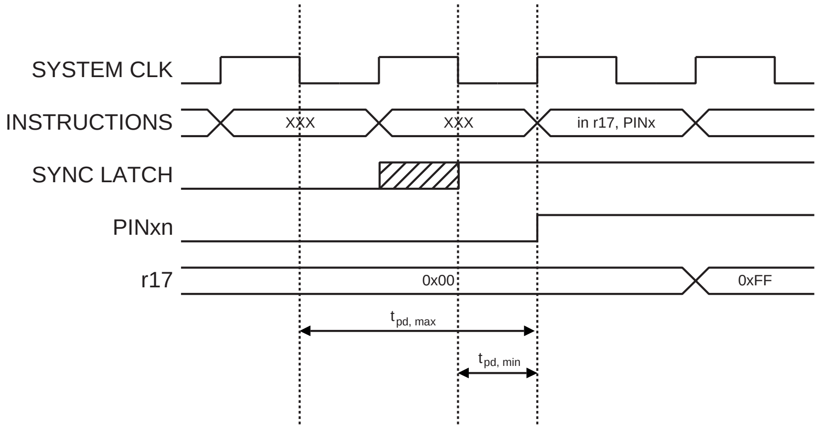
Table 10-1. Port Pin Configurations

DDxn	PORTxn	PUD (in MCUCR)	I/O	Pull-up	Comment
0	0	X	Input	No	Tri-state (Hi-Z)
0	1	0	Input	Yes	Pxn will source current if ext. pulled low
0	1	1	Input	No	Tri-state (Hi-Z)
1	0	X	Output	No	Output Low (Sink)
1	1	X	Output	No	Output High (Source)

10.1.4 Reading the Pin Value

Independent of the setting of Data Direction bit DDxn, the port pin can be read through the PINxn Register bit. As shown in Figure 10-2 on page 56, the PINxn Register bit and the preceding latch constitute a synchronizer. This is needed to avoid metastability if the physical pin changes value near the edge of the internal clock, but it also introduces a delay. Figure 10-3 shows a timing diagram of the synchronization when reading an externally applied pin value. The maximum and minimum propagation delays are denoted $t_{pd,max}$ and $t_{pd,min}$ respectively.

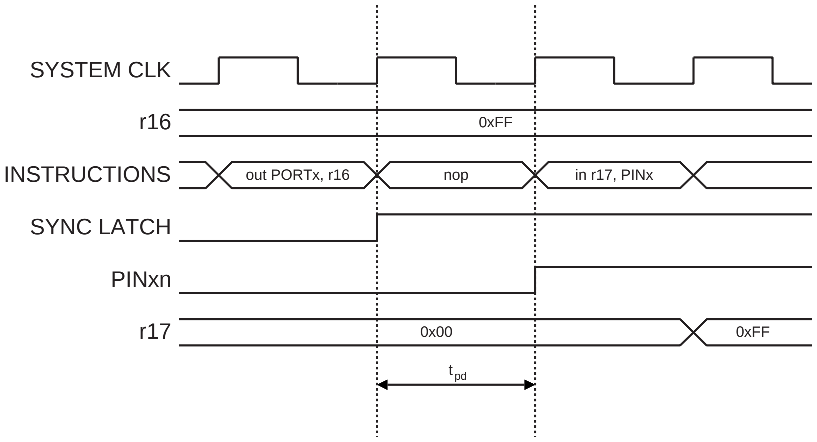
Figure 10-3. Synchronization when Reading an Externally Applied Pin value



Consider the clock period starting shortly after the first falling edge of the system clock. The latch is closed when the clock is low, and goes transparent when the clock is high, as indicated by the shaded region of the "SYNC LATCH" signal. The signal value is latched when the system clock goes low. It is clocked into the PINxn Register at the succeeding positive clock edge. As indicated by the two arrows tpd,max and tpd,min, a single signal transition on the pin will be delayed between ½ and 1½ system clock period depending upon the time of assertion.

When reading back a software assigned pin value, a nop instruction must be inserted as indicated in Figure 10-4 on page 58. The out instruction sets the "SYNC LATCH" signal at the positive edge of the clock. In this case, the delay tpd through the synchronizer is one system clock period.

Figure 10-4. Synchronization when Reading a Software Assigned Pin Value



10.1.5 Digital Input Enable and Sleep Modes

As shown in Figure 10-2 on page 56, the digital input signal can be clamped to ground at the input of the schmitt-trigger. The signal denoted SLEEP in the figure, is set by the MCU Sleep Controller in Power-down and Standby modes to avoid high power consumption if some input signals are left floating, or have an analog signal level close to $V_{CC}/2$.

SLEEP is overridden for port pins enabled as external interrupt pins. If the external interrupt request is not enabled, SLEEP is active also for these pins. SLEEP is also overridden by various other alternate functions as described in "Alternate Port Functions" on page 60.

If a logic high level ("one") is present on an asynchronous external interrupt pin configured as "Interrupt on Rising Edge, Falling Edge, or Any Logic Change on Pin" while the external interrupt is *not* enabled, the corresponding External Interrupt Flag will be set when resuming from the above mentioned Sleep mode, as the clamping in these sleep mode produces the requested logic change.

10.1.6 Unconnected Pins

If some pins are unused, it is recommended to ensure that these pins have a defined level. Even though most of the digital inputs are disabled in the deep sleep modes as described above, floating inputs should be avoided to reduce current consumption in all other modes where the digital inputs are enabled (Reset, Active mode and Idle mode).

The simplest method to ensure a defined level of an unused pin, is to enable the internal pull-up. In this case, the pull-up will be disabled during reset. If low power consumption during reset is important, it is recommended to use an external pull-up or pulldown. Connecting unused pins directly to V_{CC} or GND is not recommended, since this may cause excessive currents if the pin is accidentally configured as an output.

10.1.7 Program Examples

The following code example shows how to set port A pins 0 and 1 high, 2 and 3 low, and define the port pins from 4 to 7 as input with a pull-up assigned to port pin 4. The resulting pin values are read back again, but as previously discussed, a *nop* instruction is included to be able to read back the value recently assigned to some of the pins.

Assembly Code Example

```
...
; Define pull-ups and set outputs high
; Define directions for port pins
ldi   r16,(1<<PB4)|(1<<PB1)|(1<<PB0)
ldi   r17,(1<<DDB3)|(1<<DDB2)|(1<<DDB1)|(1<<DDB0)
out   PORTB,r16
out   DDRB,r17
; Insert nop for synchronization
nop
; Read port pins
in    r16,PINB
...
```

Note: Two temporary registers are used to minimize the time from pull-ups are set on pins 0, 1 and 4, until the direction bits are correctly set, defining bit 2 and 3 as low and redefining bits 0 and 1 as strong high drivers.

C Code Example[1]

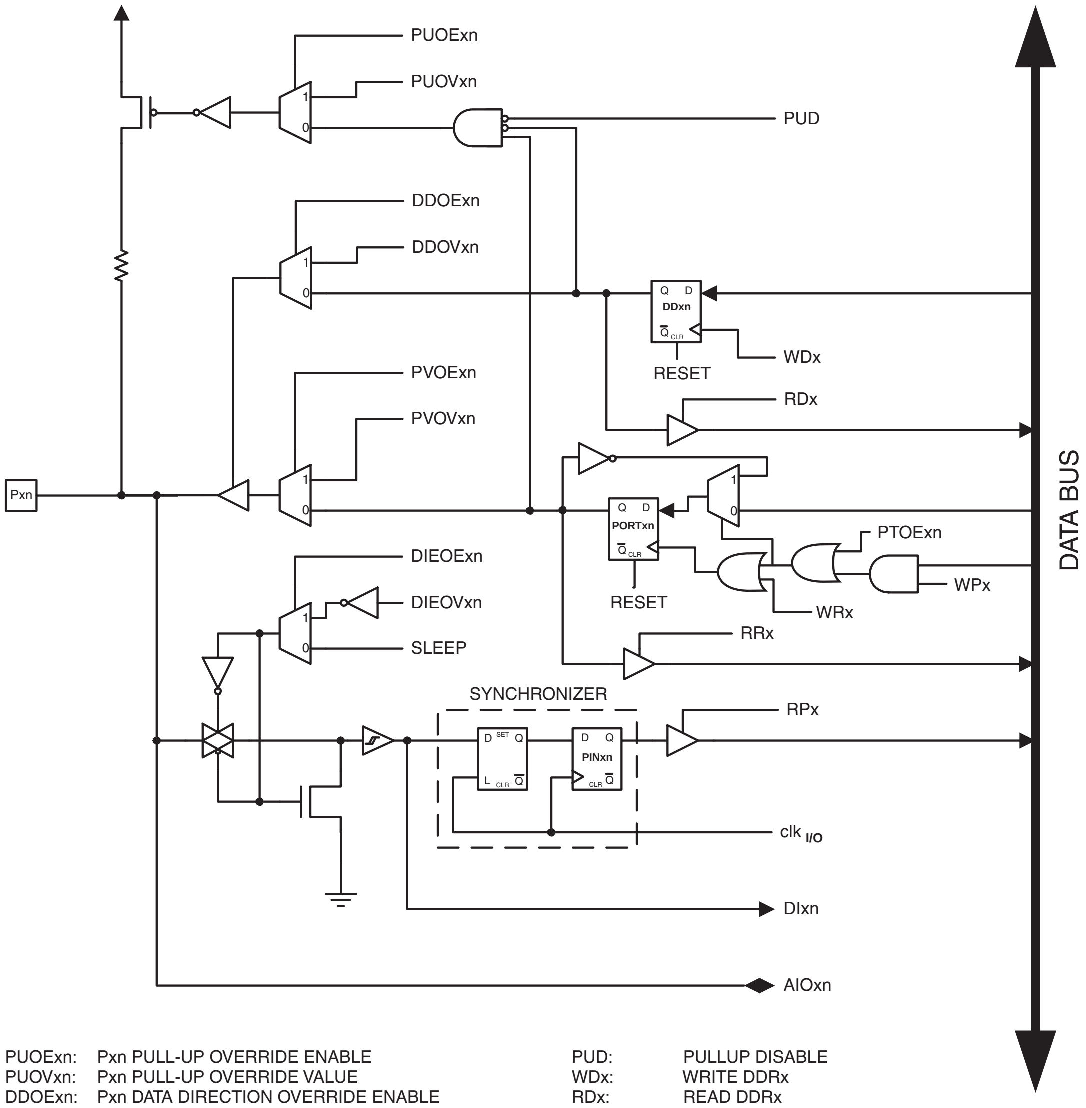
```
unsigned char i;
  ...
  /* Define pull-ups and set outputs high */
  /* Define directions for port pins */
  PORTB = (1<<PB4)|(1<<PB1)|(1<<PB0);
  DDRB = (1<<DDB3)|(1<<DDB2)|(1<<DDB1)|(1<<DDB0);
  /* Insert nop for synchronization*/
  _NOP();
  /* Read port pins */
  i = PINB;
  ...
```

Note: 1. See "Code Examples" on page 7.

10.2 Alternate Port Functions

Most port pins have alternate functions in addition to being general digital I/Os. In Figure 10-5 below is shown how the port pin control signals from the simplified Figure 10-2 on page 56 can be overridden by alternate functions.

Figure 10-5. Alternate Port Functions[1]



PUOExn:	Pxn PULL-UP OVERRIDE ENABLE
PUOVxn:	Pxn PULL-UP OVERRIDE VALUE
DDOExn:	Pxn DATA DIRECTION OVERRIDE ENABLE
DDOVxn:	Pxn DATA DIRECTION OVERRIDE VALUE
PVOExn:	Pxn PORT VALUE OVERRIDE ENABLE
PVOVxn:	Pxn PORT VALUE OVERRIDE VALUE
DIEOExn:	Pxn DIGITAL INPUT-ENABLE OVERRIDE ENABLE
DIEOVxn:	Pxn DIGITAL INPUT-ENABLE OVERRIDE VALUE
SLEEP:	SLEEP CONTROL
PTOExn:	Pxn, PORT TOGGLE OVERRIDE ENABLE

PUD:	PULLUP DISABLE
WDx:	WRITE DDRx
RDx:	READ DDRx
RRx:	READ PORTx REGISTER
WRx:	WRITE PORTx
RPx:	READ PORTx PIN
WPx:	WRITE PINx
$clk_{I/O}$:	I/O CLOCK
DIxn:	DIGITAL INPUT PIN n ON PORTx
AIOxn:	ANALOG INPUT/OUTPUT PIN n ON PORTx

Note: 1. WRx, WPx, WDx, RRx, RPx, and RDx are common to all pins within the same port. $clk_{I/O}$, SLEEP, and PUD are common to all ports. All other signals are unique for each pin.

The illustration in the figure above serves as a generic description applicable to all port pins in the AVR microcontroller family. Some overriding signals may not be present in all port pins.

Table 10-2 summarizes the function of the overriding signals. The pin and port indexes from Figure 10-5 are not shown in the succeeding tables. The overriding signals are generated internally in the modules having the alternate function.

Table 10-2. Generic Description of Overriding Signals for Alternate Functions

Signal Name	Full Name	Description
PUOE	Pull-up Override Enable	If this signal is set, the pull-up enable is controlled by the PUOV signal. If this signal is cleared, the pull-up is enabled when {DDxn, PORTxn, PUD} = 0b010.
PUOV	Pull-up Override Value	If PUOE is set, the pull-up is enabled/disabled when PUOV is set/cleared, regardless of the setting of the DDxn, PORTxn, and PUD Register bits.
DDOE	Data Direction Override Enable	If this signal is set, the Output Driver Enable is controlled by the DDOV signal. If this signal is cleared, the Output driver is enabled by the DDxn Register bit.
DDOV	Data Direction Override Value	If DDOE is set, the Output Driver is enabled/disabled when DDOV is set/cleared, regardless of the setting of the DDxn Register bit.
PVOE	Port Value Override Enable	If this signal is set and the Output Driver is enabled, the port value is controlled by the PVOV signal. If PVOE is cleared, and the Output Driver is enabled, the port Value is controlled by the PORTxn Register bit.
PVOV	Port Value Override Value	If PVOE is set, the port value is set to PVOV, regardless of the setting of the PORTxn Register bit.
PTOE	Port Toggle Override Enable	If PTOE is set, the PORTxn Register bit is inverted.
DIEOE	Digital Input Enable Override Enable	If this bit is set, the Digital Input Enable is controlled by the DIEOV signal. If this signal is cleared, the Digital Input Enable is determined by MCU state (Normal mode, sleep mode).
DIEOV	Digital Input Enable Override Value	If DIEOE is set, the Digital Input is enabled/disabled when DIEOV is set/cleared, regardless of the MCU state (Normal mode, sleep mode).
DI	Digital Input	This is the Digital Input to alternate functions. In the figure, the signal is connected to the output of the schmitt-trigger but before the synchronizer. Unless the Digital Input is used as a clock source, the module with the alternate function will use its own synchronizer.
AIO	Analog Input/Output	This is the Analog Input/Output to/from alternate functions. The signal is connected directly to the pad, and can be used bi-directionally.

The following subsections shortly describe the alternate functions for each port, and relate the overriding signals to the alternate function. Refer to the alternate function description for further details.

10.2.1 Alternate Functions of Port A

The Port A pins with alternate function are shown in Table 10-3.

Table 10-3. Port A Pins Alternate Functions

Port Pin	Alternate Function
PA0	XTAL1: Crystal Oscillator Input CLKI: External Clock Input PCINT8: Pin Change Interrupt 1, Source 8
PA1	XTAL2: Crystal Oscillator Output PCINT9: Pin Change Interrupt 1, Source 9
PA2	$\overline{\text{RESET}}$: Reset pin dW: debugWire I/O PCINT10:Pin Change Interrupt 1, Source 10

- **Port A, Bit 0 – XTAL1/CLKI/PCINT8**
  - XTAL1: Chip Clock Oscillator pin 1. Used for all chip clock sources except internal calibratable RC oscillator. When used as a clock pin, the pin can not be used as an I/O pin. When using internal calibratable RC Oscillator as a chip clock source, PA0 serves as an ordinary I/O pin.
  - CLKI: Clock Input from an external clock source, see "External Clock" on page 27.
  - PCINT8: Pin Change Interrupt source 8. The PA0 pin can serve as an external interrupt source for pin change interrupt 1.

- **Port A, Bit 1 – XTAL2/PCINT9**
  - XTAL2: Chip Clock Oscillator pin 2. Used as clock pin for all chip clock sources except internal calibratable RC Oscillator and external clock. When used as a clock pin, the pin can not be used as an I/O pin. When using internal calibratable RC Oscillator or External clock as a Chip clock sources, PA1 serves as an ordinary I/O pin.
  - PCINT9: Pin Change Interrupt source 9. The PA1 pin can serve as an external interrupt source for pin change interrupt 1.

- **Port A, Bit 2 – $\overline{\text{RESET}}$/dW/PCINT10**
  - $\overline{\text{RESET}}$: External $\overline{\text{Reset}}$ input is active low and enabled by unprogramming ("1") the RSTDISBL Fuse. Pullup is activated and output driver and digital input are deactivated when the pin is used as the $\overline{\text{RESET}}$ pin.
  - dW: When the debugWIRE Enable (DWEN) Fuse is programmed and Lock bits are unprogrammed, the debugWIRE system within the target device is activated. The RESET port pin is configured as a wire-AND (open-drain) bi-directional I/O pin with pull-up enabled and becomes the communication gateway between target and emulator.
  - PCINT10: Pin Change Interrupt source 10. The PA2 pin can serve as an external interrupt source for pin change interrupt 1.

Table 10-4 relates the alternate functions of Port A to the overriding signals shown in Figure 10-5 on page 60.

Table 10-4. Overriding Signals for Alternate Functions in PA2..PA0

Signal Name	PA2/RESET/dW/PCINT10	PA1/XTAL2/PCINT9	PA0/XTAL1/PCINT8
PUOE	$\overline{\text{RSTDISBL}}$[1] + DEBUGWIRE_ENABLE[2]	EXT_OSC[3]	EXT_CLOCK[4] + EXT_OSC[3]
PUOV	1	0	0
DDOE	$\overline{\text{RSTDISBL}}$[1] + DEBUGWIRE_ENABLE[2]	EXT_OSC[3]	EXT_CLOCK[4] + EXT_OSC[3]
DDOV	DEBUGWIRE_ENABLE[2] • $\overline{\text{debugWire Transmit}}$	0	0
PVOE	$\overline{\text{RSTDISBL}}$[1] + DEBUGWIRE_ENABLE[2]	EXT_OSC[3]	EXT_CLOCK[4] + EXT_OSC[3]
PVOV	0	0	0
PTOE	0	0	0
DIEOE	$\overline{\text{RSTDISBL}}$[1] + DEBUGWIRE_ENABLE[2] + PCINT10 • PCIE1	EXT_OSC[3] + PCINT9 • PCIE1	EXT_CLOCK[4] + EXT_OSC[3] + (PCINT8 • PCIE1)
DIEOV	DEBUGWIRE_ENABLE[2] + ($\overline{\text{RSTDISBL}}$[1] • PCINT10 • PCIE1)	$\overline{\text{EXT\_OSC}}$[3] + PCINT9 • PCIE1	(EXT_CLOCK[4] • $\overline{\text{PWR\_DOWN}}$) + $\overline{\text{(EXT\_CLOCK}}$[4] • $\overline{\text{EXT\_OSC}}$[3] • PCINT8 • PCIE1)
DI	dW/PCINT10 Input	PCINT9 Input	CLKI/PCINT8 Input
AIO		XTAL2	XTAL1

Notes:
1. RSTDISBL is 1 when the fuse is “0” (Programmed).
2. DebugWIRE is enabled when DWEN Fuse is programmed and Lock bits are unprogrammed.
3. EXT_OSC = crystal oscillator or low frequency crystal oscillator is selected as system clock.
4. EXT_CLOCK = external closk is selected as system clock.

10.2.2 Alternate Functions of Port B

The Port B pins with alternate function are shown in Table 10-5.

Table 10-5. Port B Pins Alternate Functions

Port Pin	Alternate Function
PB0	AIN0: Analog Comparator, Positive Input PCINT0:Pin Change Interrupt 0, Source 0
PB1	AIN1: Analog Comparator, Negative Input PCINT1: Pin Change Interrupt 0, Source 1
PB2	OC0A:: Timer/Counter0 Compare Match AOutput PCINT2: Pin Change Interrupt 0, Source 2
PB3	OC1A: Timer/Counter1 Compare Match A Output PCINT3: Pin Change Interrupt 0, Source 3

Table 10-5. Port B Pins Alternate Functions

Port Pin	Alternate Function
PB4	OC1B: Timer/Counter1 Compare Match B Output PCINT4: Pin Change Interrupt 0, Source 4
PB5	DI: USI Data Input (Three Wire Mode) SDA: USI Data Input (Two Wire Mode) PCINT5: Pin Change Interrupt 0, Source 5
PB6	DO: USI Data Output (Three Wire Mode) PCINT6: Pin Change Interrupt 0, Source 6
PB7	USCK: USI Clock (Three Wire Mode) SCL : USI Clock (Two Wire Mode) PCINT7: Pin Change Interrupt 0, Source 7

- **Port B, Bit 0 – AIN0/PCINT0**
  - AIN0: Analog Comparator Positive input. Configure the port pin as input with the internal pull-up switched off to avoid the digital port function from interfering with the function of the Analog Comparator.
  - PCINT0: Pin Change Interrupt Source 0. The PB0 pin can serve as an external interrupt source for pin change interrupt 0.

- **Port B, Bit 1 – AIN1/PCINT1**
  - AIN1: Analog Comparator Negative input. Configure the port pin as input with the internal pull-up switched off to avoid the digital port function from interfering with the function of the analog comparator.
  - PCINT1: Pin Change Interrupt Source 1. The PB1 pin can serve as an external interrupt source for pin change interrupt 0.

- **Port B, Bit 2 – OC0A/PCINT2**
  - OC0A: Output Compare Match A output. The PB2 pin can serve as an external output for the Timer/Counter0 Output Compare A. The pin has to be configured as an output (DDB2 set (one)) to serve this function. The OC0A pin is also the output pin for the PWM mode timer function.
  - PCINT2: Pin Change Interrupt Source 2. The PB2 pin can serve as an external interrupt source for pin change interrupt 0.

- **Port B, Bit 3 – OC1A/PCINT3**
  - OC1A: Output Compare Match A output: The PB3 pin can serve as an external output for the Timer/Counter1 Output Compare A. The pin has to be configured as an output (DDB3 set (one)) to serve this function. The OC1A pin is also the output pin for the PWM mode timer function.
  - PCINT3: Pin Change Interrupt Source 3: The PB3 pin can serve as an external interrupt source for pin change interrupt 0.

- **Port B, Bit 4 – OC1B/PCINT4**
  - OC1B: Output Compare Match B output: The PB4 pin can serve as an external output for the Timer/Counter1 Output Compare B. The pin has to be configured as an output (DDB4 set

(one)) to serve this function. The OC1B pin is also the output pin for the PWM mode timer function.

- PCINT4: Pin Change Interrupt Source 4. The PB4 pin can serve as an external interrupt source for pin change interrupt 0.

- **Port B, Bit 5 – DI/SDA/PCINT5**
  - DI: Three-wire mode Universal Serial Interface Data input. Three-wire mode does not override normal port functions, so pin must be configured as an input. SDA: Two-wire mode Serial Interface Data.
  - PCINT5: Pin Change Interrupt Source 5. The PB5 pin can serve as an external interrupt source for pin change interrupt 0.

- **Port B, Bit 6 – DO/PCINT6**
  - DO: Three-wire mode Universal Serial Interface Data output. Three-wire mode Data output overrides PORTB6 value and it is driven to the port when data direction bit DDB6 is set (one). However the PORTB6 bit still controls the pull-up enabling pull-up, if direction is input and PORTB6 is set (one).
  - PCINT6: Pin Change Interrupt Source 6. The PB6 pin can serve as an external interrupt source for pin change interrupt 0.

- **Port B, Bit 7 – USCK/SCL/PCINT7**
  - USCK: Three-wire mode Universal Serial Interface Clock.
  - SCL: Two-wire mode Serial Clock for USI Two-wire mode.
  - PCINT7: Pin Change Interrupt source 7. The PB7 pin can serve as an external interrupt source for pin change interrupt 0.

Table 10-6 and Table 10-7 relate the alternate functions of Port B to the overriding signals shown in Figure 10-5 on page 60. SPI MSTR INPUT and SPI SLAVE OUTPUT constitute the MISO signal, while MOSI is divided into SPI MSTR OUTPUT and SPI SLAVE INPUT.

Table 10-6. Overriding Signals for Alternate Functions in PB7..PB4

Signal Name	PB7/USCK/ SCL/PCINT7	PB6/DO/PCINT6	PB5/SDA/ DI/PCINT5	PB4/OC1B/ PCINT4
PUOE	USI_TWO_WIRE	0	0	0
PUOV	0	0	0	0
DDOE	USI_TWO_WIRE	0	USI_TWO_WIRE	0
DDOV	(USI_SCL_HOLD+ $\overline{\text{PORTB7}}$)•DDB7	0	($\overline{\text{SDA}}$ + $\overline{\text{PORTB5}}$)• DDB5	0
PVOE	USI_TWO_WIRE • DDB7	USI_THREE_WIRE	USI_TWO_WIRE • DDB5	OC1B_PVOE
PVOV	0	DO	0	0OC1B_PVOV
PTOE	USI_PTOE	0	0	0
DIEOE	(PCINT7•PCIE) +USISIE	(PCINT6•PCIE)	(PCINT5•PCIE) + USISIE	(PCINT4•PCIE)
DIEOV	1	1	1	1
DI	PCINT7 Input USCK Input SCL Input	PCINT6 Input	PCINT5 Input SDA Input DI Input	PCINT4 Input
AIO	–	–	–	–

Table 10-7. Overriding Signals for Alternate Functions in PB3..PB0

Signal Name	PB3/OC1A/ PCINT3	PB2/OC0A/ PCINT2	PB1/AIN1/ PCINT1	PB0/AIN0/ PCINT0
PUOE	0	0	0	0
PUOV	0	0	0	0
DDOE	0	0	0	0
DDOV	0	0	0	0
PVOE	OC1A_PVOE	OC0A_PVOE	0	0
PVOV	OC1A_PVOV	OC0A_PVOV	0	0
PTOE	0	0	0	0
DIEOE	(PCINT3 • PCIE)	(PCINT2 • PCIE)	(PCINT1 • PCIE)	(PCINT0 • PCIE)
DIEOV	1	1	1	1
DI	PCINT7 Input	PCINT6 Input	PCINT5 Input	PCINT4 Input
AIO	–	–	AIN1	AIN0

10.2.3 Alternate Functions of Port D

The Port D pins with alternate functions are shown in Table 10-8..

Table 10-8. Port D Pins Alternate Functions

Port Pin	Alternate Function
PD0	RXD: UART Data Receiver PCINT11:Pin Change Interrupt 2, Source 11
PD1	TXD: UART Data Transmitter PCINT12:Pin Change Interrupt 2, Source 12
PD2	INT0: External Interrupt 0 Input XCK: USART Transfer Clock CKOUT: System Clock Output PCINT13:Pin Change Interrupt 2, Source 13
PD3	INT1: External Interrupt 1 Input PCINT14:Pin Change Interrupt 2, Source 14
PD4	T0: Timer/Counter0 Clock Source PCINT15:Pin Change Interrupt 2, Source 15
PD5	OC0B: Timer/Counter0 Compare Match B output T1: Timer/Counter1 Clock Source PCINT16:Pin Change Interrupt 2, Source 16
PD6	ICPI: Timer/Counter1 Input Capture Pin PCINT17:Pin Change Interrupt 2, Source 17

- **Port D, Bit 0 – RXD/PCINT11**
  - RXD: UART Data Receiver.
  - PCINT11: Pin Change Interrupt Source 11. The PD0 pin can serve as an external interrupt source for pin change interrupt 2.
- **Port D, Bit 1 – TXD/PCINT12**
  - TXD: UART Data Transmitter.
  - PCINT12: Pin Change Interrupt Source 12. The PD1 pin can serve as an external interrupt source for pin change interrupt 2.
- **Port D, Bit 2 – INT0/XCK/CKOUT/PCINT13**
  - INT0: External Interrupt Source 0. The PD2 pin can serve as en external interrupt source to the MCU.
  - XCK: USART Transfer Clock used only by Synchronous Transfer mode.
  - CKOUT: System Clock Output.
  - PCINT13: Pin Change Interrupt Source 13. The PD2 pin can serve as an external interrupt source for pin change interrupt 2.
- **Port D, Bit 3 – INT1/PCINT14**
  - INT1: External Interrupt Source 1. The PD3 pin can serve as an external interrupt source to the MCU.
  - PCINT14: Pin Change Interrupt Source 14. The PD3 pin can serve as an external interrupt source for pin change interrupt 2.

- **Port D, Bit 4 – T0/PCINT15**
  - T0: Timer/Counter0 External Counter Clock input is enabled by setting (one) the bits CS02 and CS01 in the Timer/Counter0 Control Register (TCCR0).
  - PCINT15: Pin Change Interrupt Source 15. The PD4 pin can serve as an external interrupt source for pin change interrupt 2.

- **Port D, Bit 5 – OC0B/T1/PCINT16**
  - OC0B: Output Compare Match B output: The PD5 pin can serve as an external output for the Timer/Counter0 Output Compare B. The pin has to be configured as an output (DDD5 set (one)) to serve this function. The OC0B pin is also the output pin for the PWM mode timer function.
  - T1: Timer/Counter1 External Counter Clock input is enabled by setting (one) the bits CS02 and CS01 in the Timer/Counter1 Control Register (TCCR1).
  - PCINT16: Pin Change Interrupt Source 16. The PD5 pin can serve as an external interrupt source for pin change interrupt 2.

- **Port D, Bit 6 – ICPI/PCINT17**
  - ICPI: Timer/Counter1 Input Capture Pin. The PD6 pin can act as an Input Capture pin for Timer/Counter1.
  - PCINT17: Pin Change Interrupt Source 17. The PD6 pin can serve as an external interrupt source for pin change interrupt 2.

Table 10-9 and Table 10-10 relates the alternate functions of Port D to the overriding signals shown in Figure 10-5 on page 60.

Table 10-9. Overriding Signals for Alternate Functions PD6..PD4

Signal Name	**PD6/ICPI/PCINT17**	**PD5/OC1B/T1/PCINT16**	**PD4/T0/PCINT15**
PUOE	0	0	0
PUOV	0	0	0
DDOE	0	0	0
DDOV	0	0	0
PVOE	0	OC1B_PVOE	0
PVOV	0	OC1B_PVOV	0
PTOE	0	0	0
DIEOE	ICPI Enable + PCINT17	T1 Enable + PCINT16	T0 Enable + PCINT15
DIEOV	PCINT17	PCINT16	PCINT15
DI	ICPI Input/PCINT17	T1 Input/PCINT16	T0 Input/PCINT15
AIO	–	–	AIN1

Table 10-10. Overriding Signals for Alternate Functions in PD3..PD0

Signal Name	PD3/INT1/ PCINT14	PD2/INT0/XCK/CKOUT/ PCINT13	PD1/TXD/ PCINT12	PD0/RXD/PCINT11
PUOE	0	0	TXD_OE	RXD_OE
PUOV	0	0	0	PORTD0 • $\overline{\text{PUD}}$
DDOE	0	0	TXD_OE	RXD_EN
DDOV	0	0	1	0
PVOE	0	XCKO_PVOE	TXD_OE	0
PVOV	0	XCKO_PVOV	TXD_PVOV	0
PTOE	0	0	0	0
DIEOE	INT1 Enable + PCINT14	INT0 Enable/ XCK Input Enable/PCINT13	PCINT12	PCINT11
DIEOV	PCINT14	PCINT13	PCINT12	PCINT11
DI	INT1 Input/ PCINT14	INT0 Input/XCK Input/ PCINT13	PCINT12	RXD Input/PCINT11
AIO	–	–	–	–

10.3 Register Description

10.3.1 MCUCR – MCU Control Register

Bit	7	6	5	4	3	2	1	0	
0x35 (0x55)	PUD	SM1	SE	SM0	ISC11	ISC10	ISC01	ISC00	MCUCR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7 – PUD: Pull-up Disable**

When this bit is written to one, the pull-ups in the I/O ports are disabled even if the DDxn and PORTxn Registers are configured to enable the pull-ups ({DDxn, PORTxn} = 0b01). See "Configuring the Pin" on page 56 for more details about this feature.

10.3.2 PORTA – Port A Data Register

Bit	7	6	5	4	3	2	1	0	
0x1B (0x3B)	–	–	–	–	–	PORTA2	PORTA1	PORTA0	PORTA
Read/Write	R	R	R	R	R	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

10.3.3 DDRA – Port A Data Direction Register

Bit	7	6	5	4	3	2	1	0	
0x1A (0x3A)	–	–	–	–	–	DDA2	DDA1	DDA0	DDRA
Read/Write	R	R	R	R	R	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

10.3.4 PINA – Port A Input Pins Address

Bit	7	6	5	4	3	2	1	0	
0x19 (0x39)	–	–	–	–	–	PINA2	PINA1	PINA0	PINA
Read/Write	R	R	R	R	R	R/W	R/W	R/W	
Initial Value	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	

10.3.5 PORTB – Port B Data Register

Bit	7	6	5	4	3	2	1	0	
0x18 (0x38)	PORTB7	PORTB6	PORTB5	PORTB4	PORTB3	PORTB2	PORTB1	PORTB0	PORTB
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

10.3.6 DDRB – Port B Data Direction Register

Bit	7	6	5	4	3	2	1	0	
0x17 (0x37)	DDB7	DDB6	DDB5	DDB4	DDB3	DDB2	DDB1	DDB0	DDRB
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

10.3.7 PINB – Port B Input Pins Address

Bit	7	6	5	4	3	2	1	0	
0x16 (0x36)	PINB7	PINB6	PINB5	PINB4	PINB3	PINB2	PINB1	PINB0	PINB
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	

10.3.8 PORTD – Port D Data Register

Bit	7	6	5	4	3	2	1	0	
0x12 (0x32)	–	PORTD6	PORTD5	PORTD4	PORTD3	PORTD2	PORTD1	PORTD0	PORTD
Read/Write	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

10.3.9 DDRD – Port D Data Direction Register

Bit	7	6	5	4	3	2	1	0	
0x11 (0x31)	–	DDD6	DDD5	DDD4	DDD3	DDD2	DDD1	DDD0	DDRD
Read/Write	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

10.3.10 PIND – Port D Input Pins Address

Bit	7	6	5	4	3	2	1	0	
0x10 (0x30)	–	PIND6	PIND5	PIND4	PIND3	PIND2	PIND1	PIND0	PIND
Read/Write	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	

5.3. Hojas del Fabricante de los LEDs

ELECTRONICS, INC.
44 FARRAND STREET
BLOOMFIELD, NJ 07003
(973) 748–5089
http://www.nteinc.com

NTE3019
Light Emitting Diode (LED)
Red Diffused, 5mm

Features:

- Tapered Barrel T−1 3/4 Package
- High Intensity Red light source with various lens colors and effects
- Versatile Mounting on PC Board or Panel
- T−1 3/4 with Stand−off

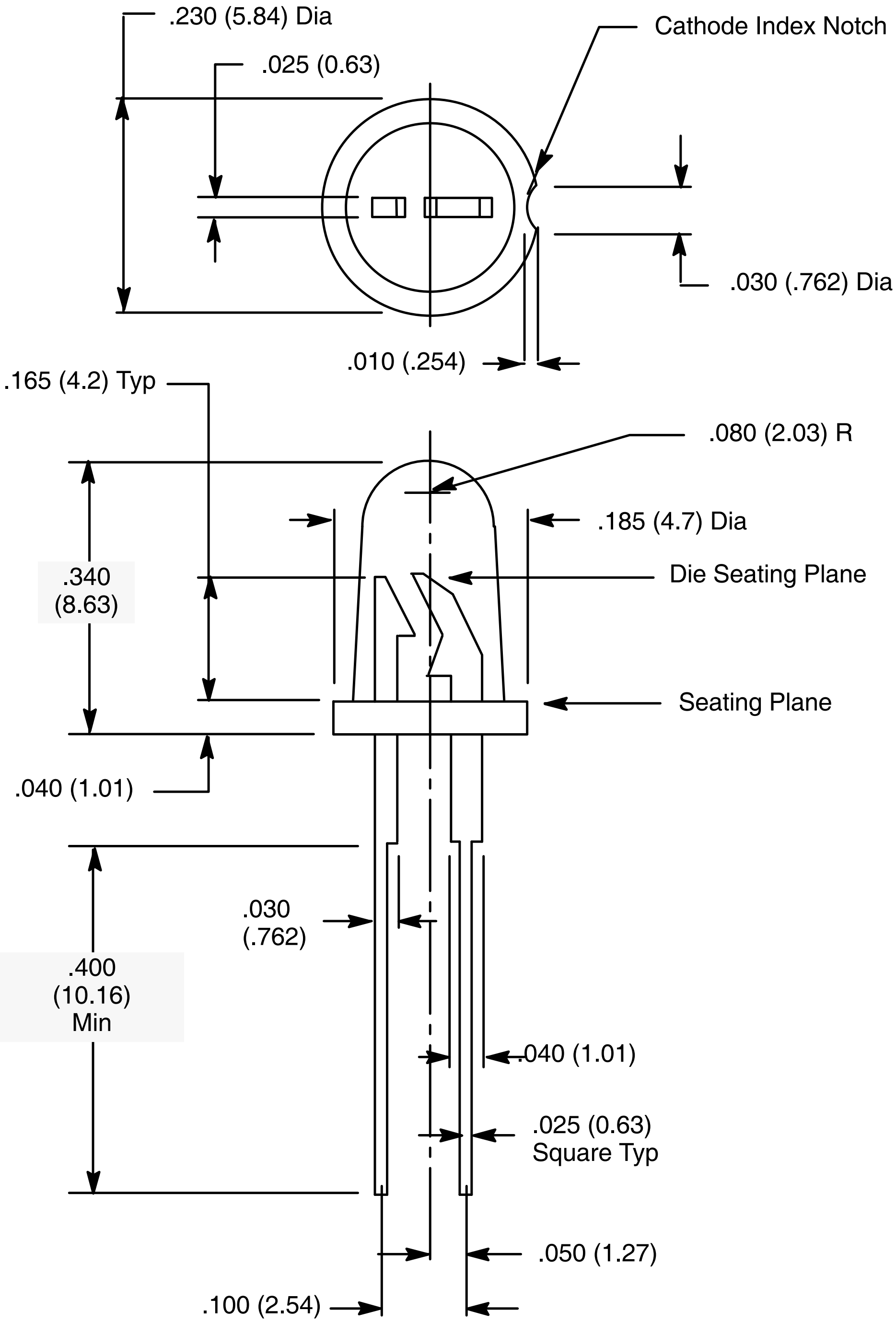
Absolute Maximum Ratings: (T_A = +25°C unless otherwise specified)

Reverse Voltage, V_R 5V
Peak Forward Current (Note 1, I_F 1A
Power Dissipation (T_A = +25°C), P_D 180mW
Derate linearly from 25°C . . . . 2mW/°C
Operating Temperature Range, T_{opr} . . . . –55° to +100°C
Storage Temperature Range, T_{stg} . . . . –55° to +100°C
Lead Temperature (During Soldering, 1/16” (1.6mm) from case, 5sec max), T_L . . . . +260°C

Note 1. Pulse Width = 1μs, 0.3% duty cycle.

Electrical Characteristics: (T_A = +25°C unless otherwise specified)

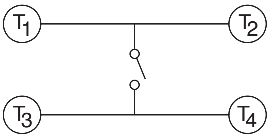
Parameter	Symbol	Test Conditions	Min	Typ	Max	Unit
Luminous Intensity	I_V	I_F = 20 mA	0.9	3.0	–	mcd
Peak Wavelength	λ_p	I_F = 20 mA	–	–	660	nm
Spectral Line Half Width	$\Delta\lambda$	I_F = 20 mA	–	20	–	nm
Forward Voltage	V_F	I_F = 20 mA	–	1.65	2.0	V
Reverse Current	I_n	V_R = 5.0V	–	–	100	λA
Reverse Voltage	λA	I_R = 100 λA	–	5.0	–	V
Capacitance	C	V = 0	–	35	–	pF
Viewing Angle	2θ1/2	Between 50% Points	–	60	–	degree
Rise Time	t_r	10% – 90% 50Ω	–	50	–	ns
Fall Time	t_f	90% – 10% 50Ω	–	50	–	ns


.230 (5.84) Dia
.025 (0.63)
Cathode Index Notch
.030 (.762) Dia
.010 (.254)
.165 (4.2) Typ
.080 (2.03) R
.185 (4.7) Dia
Die Seating Plane
.340
(8.63)
Seating Plane
.040 (1.01)
.030
(.762)
.400
(10.16)
Min
.040 (1.01)
.025 (0.63)
Square Typ
.050 (1.27)
.100 (2.54)

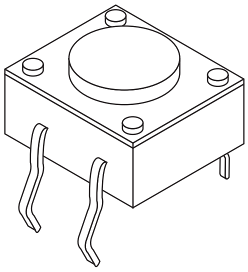
5.4. Hojas del Fabricante del botón

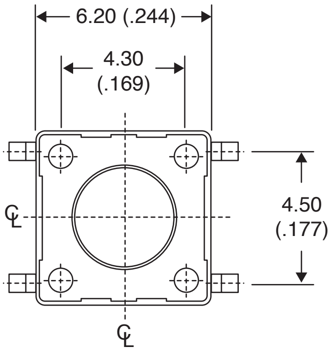
Last Update: 12/27/05 **Ref. #:** MS-100630

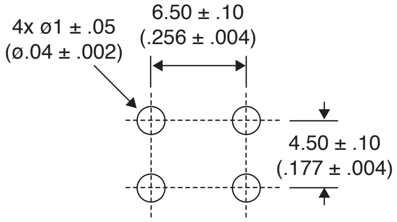
Dimensions: mm (In.)



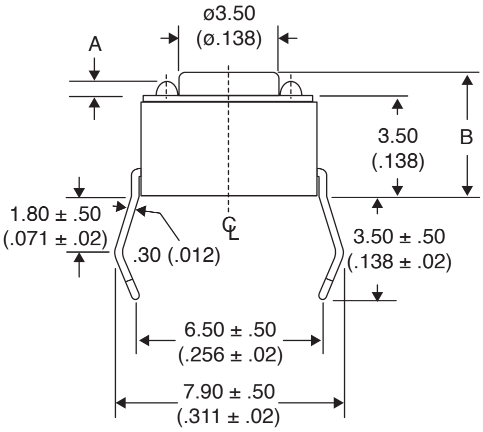
CIRCUIT DIAGRAM

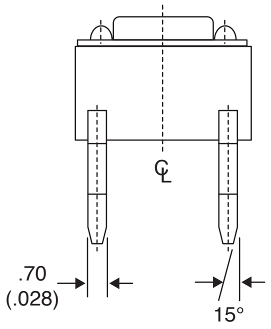






P.C.B. LAYOUT





Mechanical Specifications:

- Operating Life: 300,000 cycles min.
- Operating Temperature: -25°C to +70°C

Note:

- RoHS Compliant and process compatible with 260° solder

Electrical Specifications:

- Rating: 50mA / 12VDC
- Contact Resistance: 100mΩ max.
- Insulation Resistance: 100MΩ min.
- Dielectric Strength: 250VAC for 1 minute

Mouser Stock No.	Dimensions: mm (In)	
	A Max.	B
101-TS6111T1601-EV	0.5 (.02)	4.3 (.169)
101-TS6111T1602-EV	0.6 (.024)	5.0 (.197)
101-TS6111T1603-EV	0.6 (.024)	7.0 (.275)
101-TS6111T1606-EV	0.6 (.024)	9.5 (.374)
101-TS6111T1607-EV	0.6 (.024)	13.0 (.512)

Tactile Switches

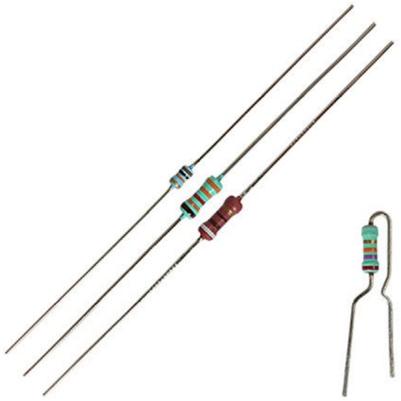
101-TS6111T1601-EV, 101-TS6111T1602-EV,
101-TS6111T1603-EV, 101-TS6111T1606-EV,
101-TS6111T1607-EV

Mountain Switch

Specifications are subject to change without notice. No liability or warranty implied by this information. Environmental compliance based on producer documentation.

5.5. Hojas del Fabricante de las resistencias

Standard Metal Film Leaded Resistors



FEATURES

- Small size (SFR16S: 0204, SFR25 / SFR25H: 0207)
- Low noise (max. 1.5 μV/V for R > 1 MΩ)
- Compatible to both lead (Pb)-free and lead containing soldering processes
- Material categorization: for definitions of compliance please see www.vishay.com/doc?99912

Pb-free
RoHS COMPLIANT
HALOGEN FREE

APPLICATIONS

- General purpose resistors

A homogeneous film of metal alloy is deposited on a high grade ceramic body. After a helical groove has been cut in the resistive layer, tinned connecting leads of electrolytic copper are welded to the end-caps.

The resistors are coated with a colored lacquer (light-blue for type SFR16S; light-green for type SFR25 and red-brown for type SFR25H) which provides electrical, mechanical, and climatic protection. The encapsulation is resistant to all cleaning solvents in accordance with IEC 60068-2-45.

TECHNICAL SPECIFICATIONS

DESCRIPTION	SFR16S	SFR25	SFR25H
DIN size	0204	0207	0207
Resistance range	1 Ω to 3 MΩ; jumper (0 Ω)	0.22 Ω to 10 MΩ; jumper (0 Ω)	0.22 Ω to 10 MΩ
Resistance tolerance	± 5 %; ± 1 %		
Temperature coefficient	± 250 ppm/K; ± 100 ppm/K		
Rated dissipation, P_{70}	0.5 W	0.4 W	0.5 W
Thermal resistance	170 K/W	200 K/W	150 K/W
Operating voltage, $U_{max.}$ AC/DC	200 V	250 V	350 V
Operating temperature range	-55 °C to 155 °C		
Permissible film temperature	155 °C		
Max. resistance change at rated dissipation $\|\Delta R/R$ max.$\|$, after 1000 h	± (2 % R + 0.05 Ω)		

Note

- R value is measured with probe distance of 24 mm ± 1 mm using 4-terminal method.