Universidad de Costa Rica

Facultad de Ingeniería
Escuela de Ingeniería Eléctrica

IE-0624
Laboratorio de Microcontroladores

Laboratorio #2: GPIOs, Timers y FSM

Gabriel Alberto Barahona Otoya B70896

Profesor:
Marco Villalta Fallas

II-Ciclo, 2022

Índice

1. Introducción

1.1. Resumen

El objetivo principal de este segundo laboratorio es estudiar el funcionamiento y todas las características que ofrece el microcontrolador AT-tiny4313 como la manipulación de GPIOs (General Purpose Input Outputs), interrupciones y timers. Además, se pone en práctica el desarrollo de máquinas de estados, diagramas de temporización y diagramas de flujo para la resolución de problemas. El laboratorio consiste en el desarrollo de un cruce de semáforos simplificado utilizando LEDs, botones y el microcontrolador mencionado. En el circuito se utilizarán al menos 6 leds, los LEDs LDPV y LDVD representan el semáforo vehicular. Por otro lado, los LEDs LDPP y LDPD corresponden a la representación de un semáforo peatonal. Los botones B1 y B2 se usarán para que los usuarios soliciten la activación de las luces peatonales para poder cruzar la calle.

- LDPV: paso de vehículos.
- LDPP: paso de peatones.
- LDVD: vehículos detenidos.
- LDPD: peatones detenidos.

1.2. Conclusiones importantes

En comparación con el microcontrolador utilizado en el primer laboratorio (PIC12F675), este es un poco más complejo, posee una mayor cantidad de pines como se muestra en la figura1. Cada pin tiene su función asociada la cual se escoge realizando la debida configuración y de acuerdo a las necesidades del proyecto. Cada una de estas características se detallarán en el aparatado de nota teórica.

2. Nota teórica

2.1. Información general del Microcontrolador

Los fabricantes de este componente lo catalogan como un microcontrolador basado en RISC AVR de 8 bits de alto rendimiento que combina ciertas característica como lo son memoria Flash de 4 KB, EEPROM de 128B, SRAM de 128B, 18 líneas de E/S y 32 registros de trabajo ambos de propósito general, una interfaz de un solo cable para depuración de chips, dos temporizadores/-contadores flexibles con modos de comparación, interrupciones internas y externas, USART serial programable, una interfaz serial universal (USI) con detector de condiciones de inicio, temporizador de vigilancia programable con oscilador interno y tres modos de ahorro de energía seleccionables por software. Este microcontrolador funciona entre 1,8 y 5,5 Volts. Es capaz de ejecutar poderosas instrucciones en un solo ciclo de reloj. Además, logra rendimientos que se acercan a un MIPS por MHz, equilibrando la velocidad de procesamiento y principalmente el consumo de energía [2].

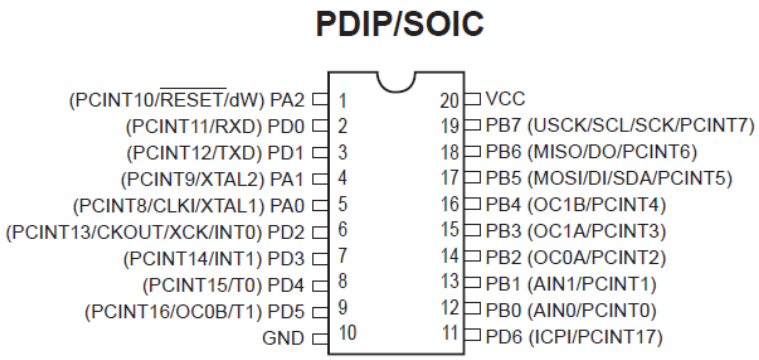


Figura 1: Diagrama de pines del AT-tiny4313

Información general [3]:

- Fabricante: Microchip
- Categoría de producto: Microcontroladores de 8 bits - MCU
- Serie: ATtiny4313
- Núcleo: AVR
- Tamaño de memoria del programa: 4 kB
- Ancho de bus de datos: 8 bit
- Frecuencia de reloj máxima: 20 MHz
- Número de entradas/salidas: 18 I/O
- Tamaño de RAM de datos: 256 B
- Voltaje de alimentación - Mín.: 1.8 V
- Voltaje de alimentación - Máx.: 5.5 V
- Temperatura de trabajo mínima: -40 C
- Temperatura de trabajo máxima: +85 C
- Tipo de Ram de datos: SRAM
- Tamaño de ROM de datos: 256 B
- Tipo de Rom de datos: EEPROM
- Tipo de interfaz: I2C, SPI, USART, USI
- Tipo de memoria de programa: Flash
- Temporizadores de vigilancia: Watchdog Timer

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

Figura 2: Características eléctricas [2]

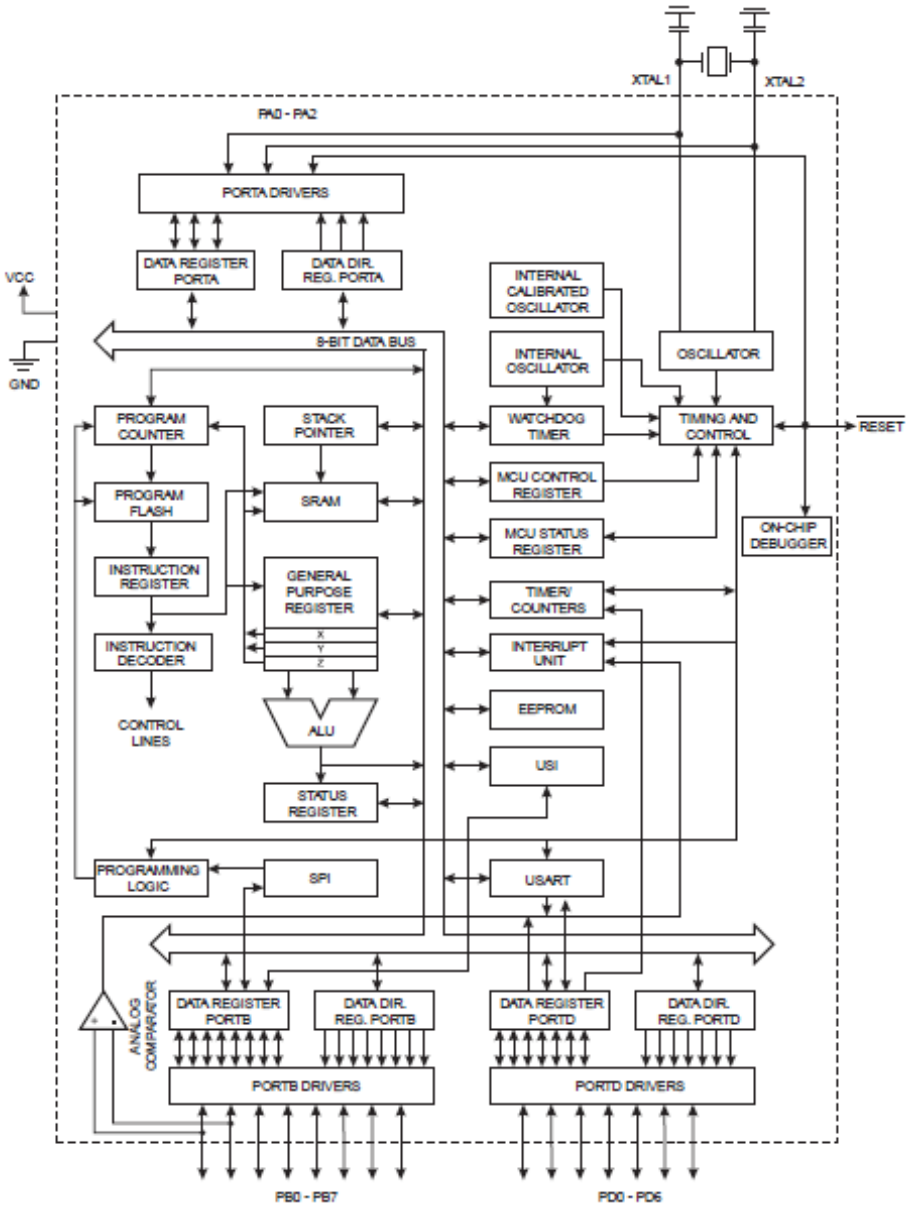


Figura 3: Diagrama de bloques del AT-tiny4313 [2]

2.2. Conceptos/temas del laboratorio: Registros

El microcontrolador en estudio consta de varios registros que se pueden operar digitalmente. A continuación, se va a explicar con detalle el funcionamiento de cada uno de los registros utilizados para la realización del laboratorio:

2.2.1. GPIOs

En el ATtiny4313, cada puerto/pin consiste de tres bits de registro: DDxn, PORTxn y PINxn (x es el puerto y n el pin). A continuación, se mencionan algunos aspectos importantes sobre GPIOs [1]:

- DDxn bit pertenece al registro DDRx: 1 para salida y 0 para entrada.
- PORTxn bit pertenece al registro PORTx. Cuando esta configurado como entrada, un 1 activa la resistencia pull-up, para apagar la resistencia se debe configurar pin como salida.
- Cuando PORTxn esta configurado como salida: si se escribe un 1 lógico, el pin se pone en alto, si se escribe un 0 lógico hace que el pin se ponga en bajo.
- Escribiendo un 1 a PINxn cambia el valor de PORTxn independiente del valor en DDRxn.
- Independiente de la configuración de DDxn, el pin se puede leer en el bit PINxn.
- Cuando se leen por interrupciones se debe configurar el registro PCMSKn y GIMSK.

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	

Figura 4: Registros de GPIOs

2.2.2. Registros de Interrupciones

En lugar de realizar polling para ver si existen nuevos datos, es más eficiente que el periférico indique la disposición de nuevos datos. Las interrupciones son definidas como notificaciones que se le realizan al CPU sobre el suceso de un evento. Pueden ser disparados por varios eventos o perifericos de entrada o salida, ADC, software, timers, etc. Es importante tener claro que estas debe ser atendidas en rutinas cortas, por lo que se utiliza normalmente para cambiar una variable o algún bit de un registro, pero nunca se debe realizar un loop ya sea de for o while en una subrutina de interrupción. Cuando una interrupción se activa, el modo de operación del programa se detiene y se guarda su estado. Inmediatamente se ejecuta una ISR (Interrupt service routine). Luego el procesador se encarga de revisar el espacio de memoria de los vectores de interrupción (registros) [1].

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

Figura 5: Vectores de interrupción y prioridades

Observando la figura1, los GPIOs de interrupciones van del PCINT17 al PCINT0. PCIE2 se dispara si PCINT17 a PCINT11 cambian. PCIE1 se dispara si PCINT10 a PCINT8 cambian. PCIE0 se dispara si PCINT7 a PCINT0 cambian. INT0 y INT1 se disparan por flanco postivo, negativo o un nivel bajo dependiendo su configuración. Si se quieren usar habilitar la interrupción global (Usar función sei()) [1].

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	

Figura 6: Registro MCUR

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	

Figura 7: Registro GIMSK

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	

Figura 8: Registro PCMSK2

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	

Figura 9: Registro PCMSK1

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	

Figura 10: Registro PCMSK0

Para mayor detalles, consultar la hoja del fabricante que se encuentra en anexos.

2.2.3. Timers

Los timers permiten la medición del intervalo de tiempo o contar eventos internos o externos. Pueden generar interrupciones y la velocidad de conteo en función de la fuente de reloj y la configuración de escala denominado prescaler. El prescaler es un circuito contador que se usa para disminuir una señal eléctrica de alta frecuencia a una de frecuencia mucho menor por medio de una división con valores de potencias de 2. En el AT-tiny4313 el prescaler para el Timer 0 se configura con el registro TCCR0B, con los Bits 2:0. En El ATtiny4313 se puede escoger un reloj externo, utilizar oscilador interno RC a 4 Mhz u oscilador interno RC a 8 Mhz, el cual es el que viene configurado por defecto [1].

Normalmente se utilizan los registros TCCR0A, TCCR0B, OCR0A, OCR0B, TIMSK y TIFR para operarlo y son los que se muestran a continuación:

11.9.1 **TCCR0A – Timer/Counter Control Register A**

Bit	7	6	5	4	3	2	1	0	
0x30 (0x50)	COM0A1	COM0A0	COM0B1	COM0B0	–	–	WGM01	WGM00	TCCR0A
Read/Write	R/W	R/W	R/W	R/W	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Figura 11: Registro TCCR0A

11.9.2 TCCR0B – Timer/Counter Control Register B

Bit	7	6	5	4	3	2	1	0	
0x33 (0x53)	FOC0A	FOC0B	–	–	WGM02	CS02	CS01	CS00	TCCR0B
Read/Write	W	W	R	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Figura 12: Registro TCCR0B

11.9.4 OCR0A – Output Compare Register A

Bit	7	6	5	4	3	2	1	0	
0x36 (0x56)	OCR0A[7:0]								OCR0A
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 13: Registro OCR0A

11.9.5 OCR0B – Output Compare Register B

Bit	7	6	5	4	3	2	1	0	
0x3C (0x5C)	OCR0B[7:0]								OCR0B
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 14: Registro OCR0B

11.9.6 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 15: Registro TIMSK

11.9.7 TIFR – Timer/Counter Interrupt Flag Register

Bit	7	6	5	4	3	2	1	0	
0x38 (0x58)	TOV1	OCF1A	OCF1B	–	ICF1	OCF0B	TOV0	OCF0A	TIFR
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 16: Registro TIFR

Para mayor detalles, consultar la hoja del fabricante que se encuentra en anexos.

2.2.4. Máquinas de estados

Permiten la creación de un sistema controlado por eventos que puede cambiar la respuesta a entradas asociadas a un estado específico. Poseen un número finito de estados para un sistema determinado. Las ventajas que ofrece el uso de máquinas de estados hacen que el código tenga un grado de mayor eficiencia, que sea más sencillo el proceso de depuración y beneficia la organización del flujo de programa. Además, incentivan al programador a utilizar buenas técnicas de diseño de firmware. Su estructura consta de entradas, salidas y estados conocidos. Las entradas hacen referencia a cualquier evento que necesita el sistema para realizar un cambio en su comportamiento. Las salidas corresponden a acciones necesarias que el sistema debe hacer como respuesta a una

entrada. Por último, los estados indican la acción que realiza el sistema cuando un evento sucede. Existen varias formas de programar una máquina de estados, se pueden usar instrucciones if/else, switch/cases, structs/enums, lookup tables o punteros a funciones [1].

2.3. Periféricos

2.3.1. LEDs

Se define como un componente electrónico semiconductor que tiene la capacidad de emitir luz cuando es atravesado por una corriente funcionando en polarización directa. Al igual que otros tipos de diodos, los LEDs poseen dos terminales de conexión, el ánodo (más largo) y el cátodo (más corto) [6].

Figura 17: LEDs

2.3.2. Botones

Son dispositivos que son activados al ser pulsados con el dedo. La pulsación permite el flujo de corriente lo cual permite su accionar. En el momento que se deja de presionar, vuelve a su modo de espera. Normalmente tienen 2 terminales, pueden operar en modo normalmente abierto o normalmente cerrado. Internamente, posee una lámina conductora que realiza un contacto con los dos terminales al presionar el botón [5].

Figura 18: Botón

2.4. Diseño del circuito

A continuación se muestra el diseño del circuito. Su funcionamiento se explica más a detalle en el apartado de análisis de resultados.

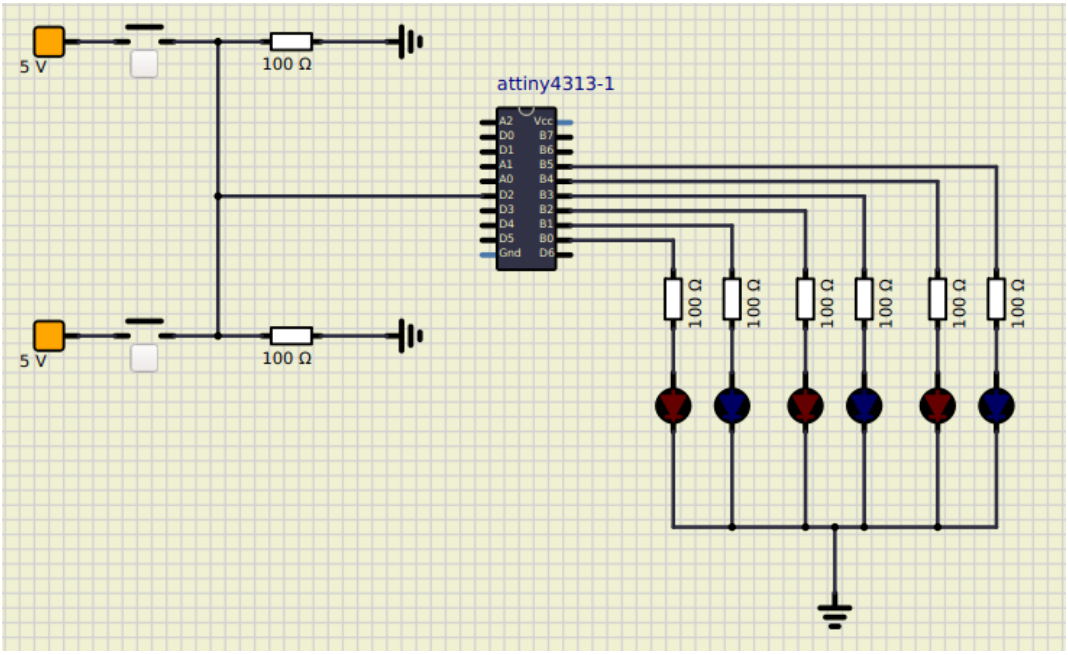


Figura 19: Circuito en SimulIDE

2.5. Lista de componentes y precios

Los precios fueron extraídos de la página web de Microchip [2] y de la tienda Teltron [4] ubicada en San José.

- Componente: AT-tiny 4313 / Precio: 1-24 $1.68
- Componente: 3 LEDs verdes / Precio: ¢237
- Componente: 3 LEDs rojos / Precio: ¢237
- Componente: 2 botones / Precio: ¢100

3. Desarrollo/Análisis de resultados

3.1. Programa

El código del laboratorio se puede encontrar en el apartado de Anexos. En esta sección se va a explicar el funcionamiento de este y cada una de las partes que lo conforman. Para la realización de este laboratorio se puso en práctica una de las funciones que ofrece el AT-tiny4313 que es el de las interrupciones, específicamente las externas, que son interrupciones disparadas por elementos externos, como puede ser que se de un evento de presionar un botón. En el código primero se observan las rutinas cortas de interrupción. Luego se configuró cada uno de los pines como salidas para encender los LEDs con el registro DDRB. Luego se llama a la función sei() para habilitar las

interrupciones. Después, se configuró el registro GIMSK, ya que este contiene los bits de activación de INT0 e INT1 que fueron utilizados en el código. Si se les configura un 1, estos habilitan las interrupciones externas de los pines del microcontrolador. El registró MCUCR, se utilizó para configurar el punto en el que debe activarse la interrupción externa, en este caso sucede cuando hay un flanco negativo.

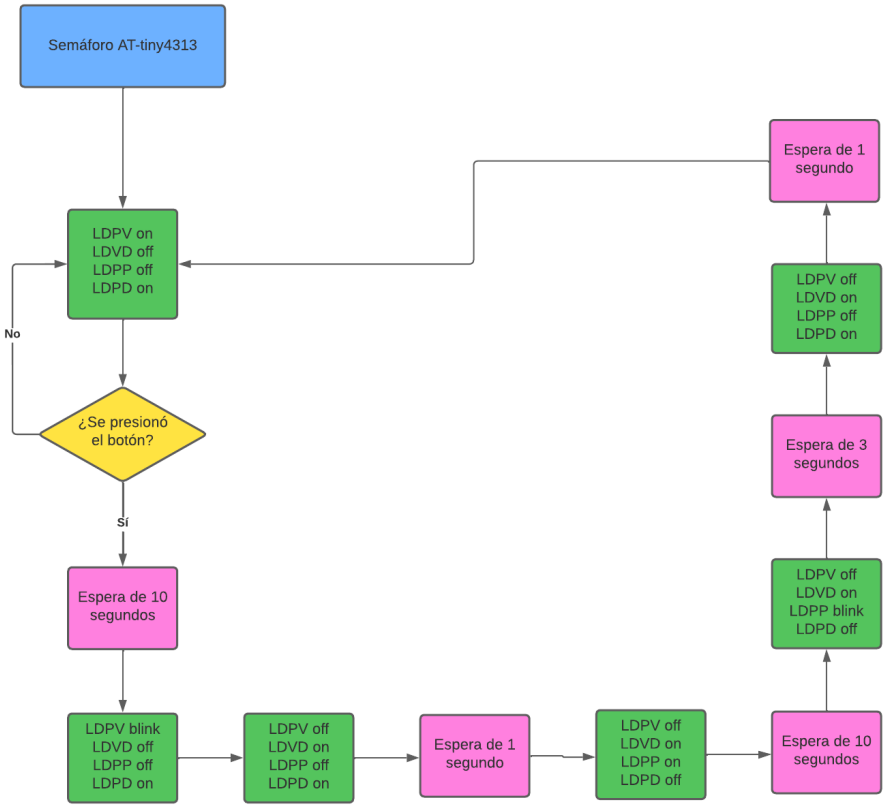


Figura 20: Diagrama de flujo

3.2. Componentes

En esta sección se explicará a detalle la razón por la cual se están utilizando cada uno de los componentes. El elemento principal es el microcontrolador AT-tiny4313 que es el que se encarga del funcionamiento del circuito en su totalidad. A este se le adaptaron 2 botones con una resistencia de pull down que hacen referencia al semáforo peatonal. También, el circuito cuenta con 6 LEDs, es importante tener claro que los primeros dos LEDs son para el semáforo vehicular y los otros cuatro son para los dos semáforos peatonales.

3.3. Demostración de funcionamiento

Para iniciar con la simulación se presiona el botón de Power Circuit. Por defecto, el circuito va a iniciar en su estado de circulación de vehículos, es decir, con LDPV en alto (semáforo vehicular en verde y el LDVD apagado) y con LDPD en alto también (semáforo peatonal en rojo y el LDPP apagado) . Al presionar el botón, se le envía una señal al microcontrolador y se pasa al siguiente

estado que es el de paso de peatones, es decir, con LDPP en alto (semáforo peatonal en verde y el LDPD apagado) y con LDVD en alto (semáforo vehicular en rojo y el LDPV apagado).

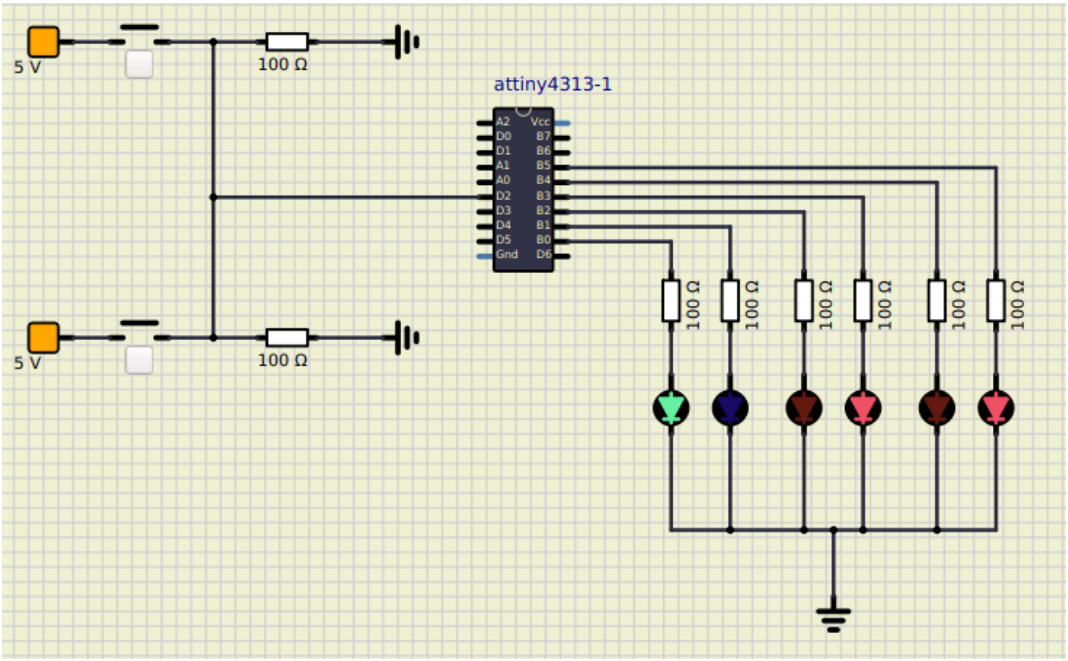


Figura 21: Paso vehicular

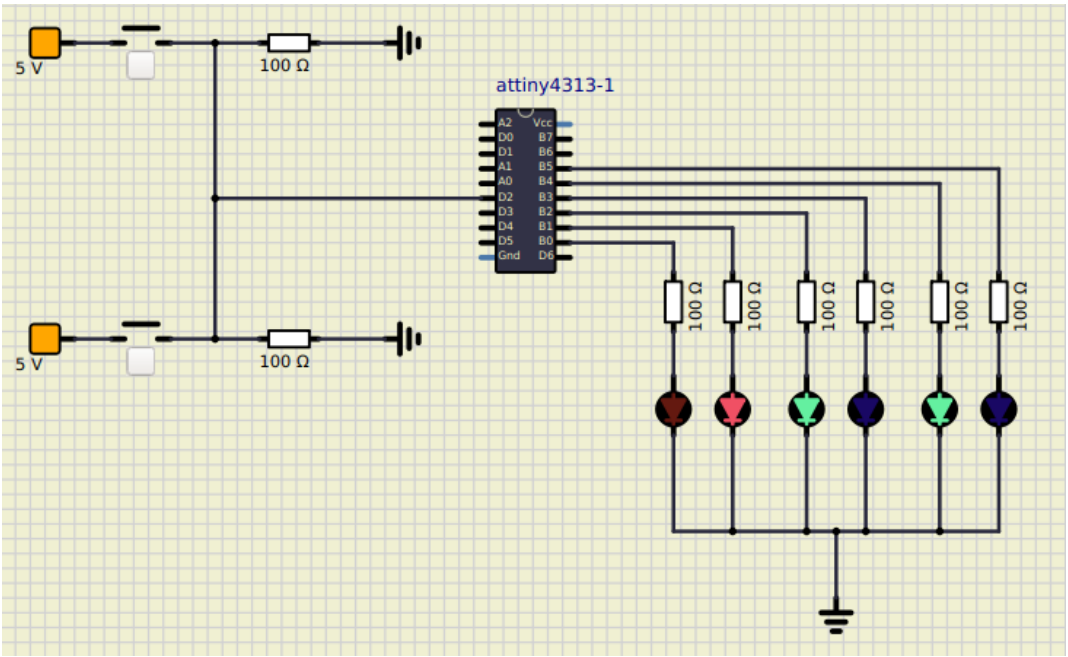


Figura 22: Paso peatonal

4. Conclusiones y recomendaciones

- Antes de iniciar a trabajar con el AT-tiny4313 es importante entender el funcionamiento de los interrupts y los timers internos que este ofrece.
- Existen otros tipos de microcontroladores que pueden facilitar la tarea de crear un sistema de semáforos.
- Antes de iniciar con el uso del circuito es importante encenderlo y apagarlo un par de veces para que se estabilice y funcione correctamente. Esto se debe a que se está trabajando con un simulador.

Referencias

[1] Villalta, M. (2022). **GPIOs y el PIC12f629**. IE-0624 Laboratorio de Microcontroladores. Escuela de Ingeniería Eléctrica.

[2] Microchip. **ATtiny4313**. Página web: `https://www.microchip.com/en-us/product/ATtiny4313`

[3] Mouser Electronics. **Microchip Technology / Atmel ATTINY4313-SU**. Página web: `https://www.mouser.co.cr/ProductDetail/Microchip-Technology-Atmel/ATTINY4313-SU?qs=lwdSMh1%2FoYIbii0kW%252Bj%2FtQ%3D%3D`

[4] Teltron (Costa Rica), S.A. Catálogo: `https://teltroncr.com/index.php?page=products.php#close`

[5] UNIT ELECTRONICS. **Push Button 2 Pines MicroSwitch**. Página web: `https://uelectronics.com/producto/push-button-2-pines-microswitch/`

[6] Martín, J. **El Diodo (Electrónica)**. Página web: `https://books.google.co.cr/books?id=IVUpDwAAQBAJ&pg=PA83&dq=diodos+led&hl=es&sa=X&ved=2ahUKEwjakrHYwqT6AhXDQjABHT-nArcQ6AF6BAgJEAI#v=onepage&q=diodos%20led&f=false`

5. Anexos

5.1. GITHUB

El control de versiones o avances del programa del laboratorio se encuentra en el siguiente repositorio público de GITHUB:

- HTTPS: https://github.com/Gabo4x/Laboratorio-Microcontroladores.git
- SSH: git@github.com:Gabo4x/Laboratorio-Microcontroladores.git

5.2. Código del laboratorio

```
//Trabajo realizado por: Gabriel Barahona Otoya B70896
//Laboratorio de Microcontroladores IE-0624
//Laboratorio #2

#include <avr/io.h>
#include <avr/interrupt.h>

ISR (INT0_vect)
{
    PORTB ^= ~(1<<PB0) | ~(1<<PB3) | ~(1<<PB5);
}
ISR (INT1_vect)
{
    PORTB &= ~(1<<PB1) & ~(1<<PB2) & ~(1<<PB4);
}
ISR (PCINT0_vect)
{
    PORTB ^= ~(1<<PB6);
}

void config()
{
  DDRB |= (1<<PB0) | (1<<PB1) | (1<<PB2) | (1<<PB3) | (1<<PB4)| (1<<PB5);
   sei();
  GIMSK |= (1<<INT0);
  MCUCR |= (1<<ISC01);
  GIMSK |= (1<<INT1);
  MCUCR |= (1<<ISC11);
}

void pin_change_interrupt()
{
  DDRB |= (1<<PB0);
  GIMSK |= (1<<PCIE1);
  PCMSK |= (1<<PCINT1);
}

int main()
{
   config();
   pin_change_interrupt();
   while(1)
   {

   }
}
```

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

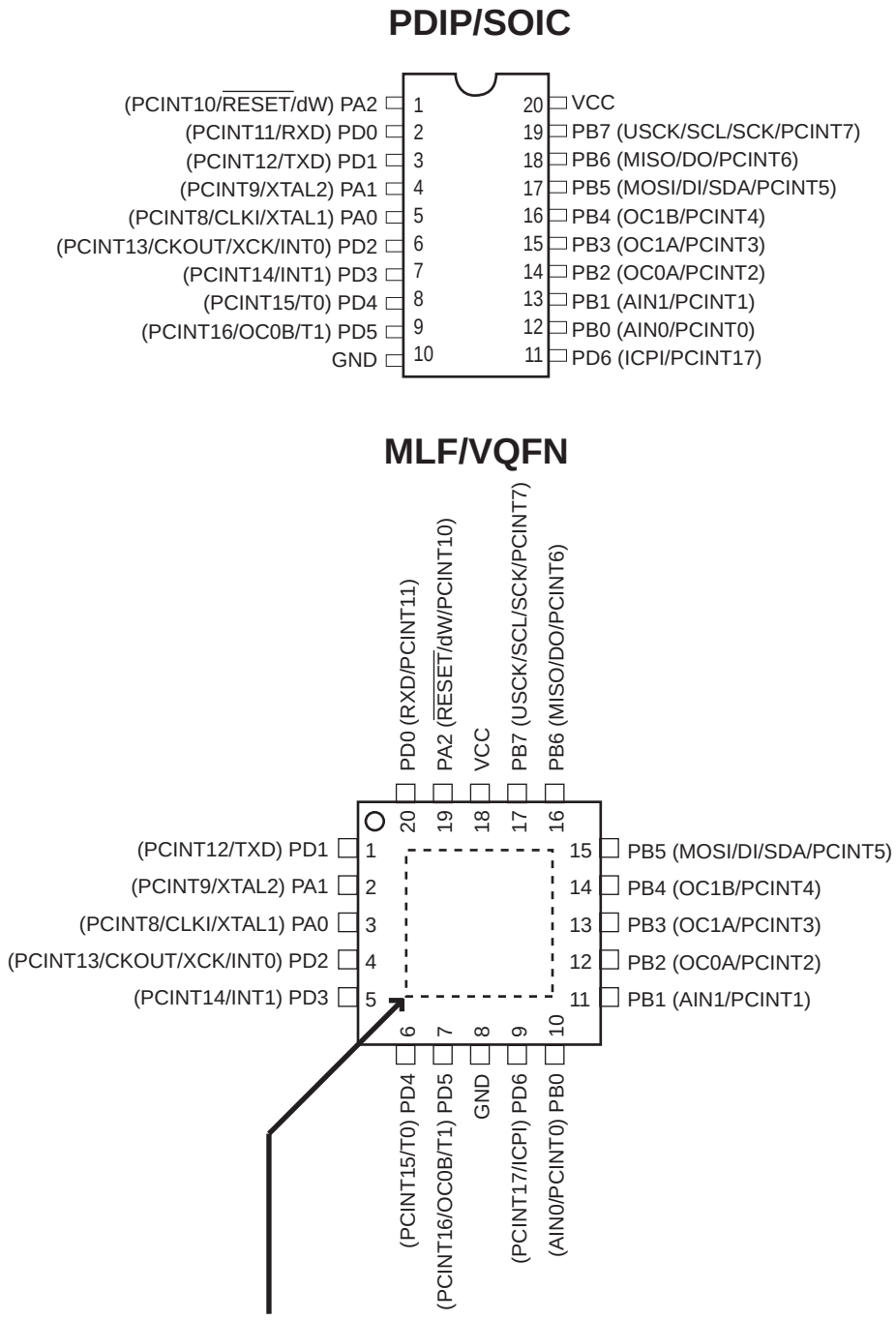
8-bit AVR® Microcontroller with 2/4K Bytes In-System Programmable Flash

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

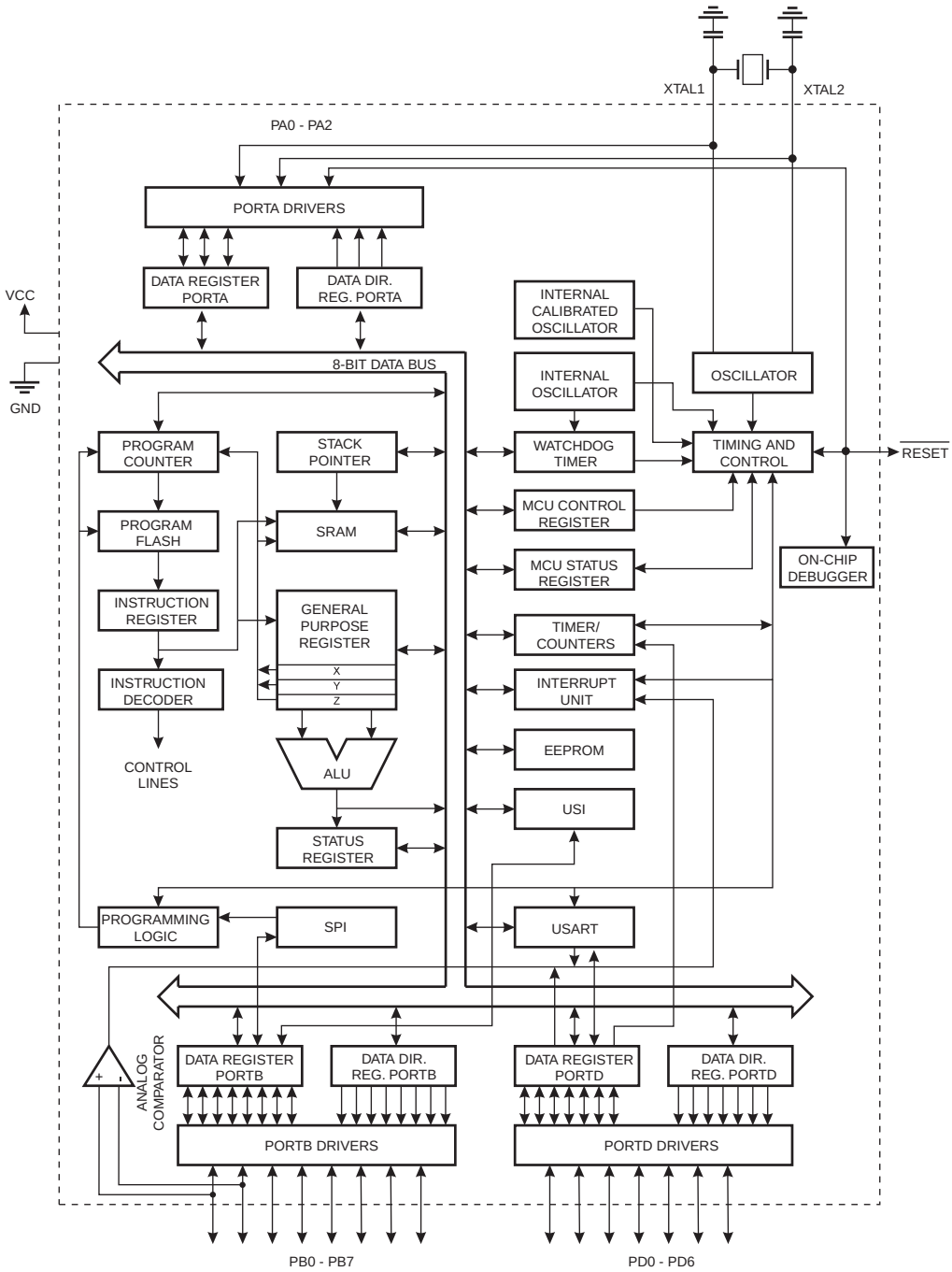
Output from the inverting Oscillator amplifier. XTAL2 is an alternate function for PA1.

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.

8.5 Register Description

8.5.1 MCUSR – MCU Status Register

The MCU Status Register provides information on which reset source caused an MCU Reset.

Bit	7	6	5	4	3	2	1	0	
0x34 (0x54)	–	–	–	–	WDRF	BORF	EXTRF	PORF	MCUSR
Read/Write	R	R	R	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	See Bit Description				

- **Bits 7..4 – Res: Reserved Bits**

These bits are reserved bits in the ATtiny2313A/4313 and will always read as zero.

- **Bit 3 – WDRF: Watchdog Reset Flag**

This bit is set if a Watchdog Reset occurs. The bit is reset by a Power-on Reset, or by writing a logic zero to the flag.

- **Bit 2 – BORF: Brown-out Reset Flag**

This bit is set if a Brown-out Reset occurs. The bit is reset by a Power-on Reset, or by writing a logic zero to the flag.

- **Bit 1 – EXTRF: External Reset Flag**

This bit is set if an External Reset occurs. The bit is reset by a Power-on Reset, or by writing a logic zero to the flag.

- **Bit 0 – PORF: Power-on Reset Flag**

This bit is set if a Power-on Reset occurs. The bit is reset only by writing a logic zero to the flag.

To make use of the Reset Flags to identify a reset condition, the user should read and then reset the MCUSR as early as possible in the program. If the register is cleared before another reset occurs, the source of the reset can be found by examining the Reset Flags.

8.5.2 WDTCSR – Watchdog Timer Control and Status Register

Bit	7	6	5	4	3	2	1	0	
0x21 (0x41)	WDIF	WDIE	WDP3	WDCE	WDE	WDP2	WDP1	WDP0	WDTCSR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	X	0	0	0	

- **Bit 7 – WDIF: Watchdog Timeout Interrupt Flag**

This bit is set when a time-out occurs in the Watchdog Timer and the Watchdog Timer is configured for interrupt. WDIF is cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, WDIF is cleared by writing a logic one to the flag. When the I-bit in SREG and WDIE are set, the Watchdog Time-out Interrupt is executed.

- **Bit 6 – WDIE: Watchdog Timeout Interrupt Enable**

When this bit is written to one, WDE is cleared, and the I-bit in the Status Register is set, the Watchdog Time-out Interrupt is enabled. In this mode the corresponding interrupt is executed instead of a reset if a timeout in the Watchdog Timer occurs.

If WDE is set, WDIE is automatically cleared by hardware when a time-out occurs. This is useful for keeping the Watchdog Reset security while using the interrupt. After the WDIE bit is cleared,

the next time-out will generate a reset. To avoid the Watchdog Reset, WDIE must be set after each interrupt.

Table 8-2. Watchdog Timer Configuration

WDE	WDIE	Watchdog Timer State	Action on Time-out
0	0	Stopped	None
0	1	Running	Interrupt
1	0	Running	Reset
1	1	Running	Interrupt

- **Bit 4 – WDCE: Watchdog Change Enable**

This bit must be set when the WDE bit is written to logic zero. Otherwise, the Watchdog will not be disabled. Once written to one, hardware will clear this bit after four clock cycles. See the description of the WDE bit for a Watchdog disable procedure. This bit must also be set when changing the prescaler bits. See “Timed Sequences for Changing the Configuration of the Watchdog Timer” on page 43.

- **Bit 3 – WDE: Watchdog Enable**

When the WDE is written to logic one, the Watchdog Timer is enabled, and if the WDE is written to logic zero, the Watchdog Timer function is disabled. WDE can only be cleared if the WDCE bit has logic level one. To disable an enabled Watchdog Timer, the following procedure must be followed:

1. In the same operation, write a logic one to WDCE and WDE. A logic one must be written to WDE even though it is set to one before the disable operation starts.
2. Within the next four clock cycles, write a logic 0 to WDE. This disables the Watchdog.

In safety level 2, it is not possible to disable the Watchdog Timer, even with the algorithm described above. See “Timed Sequences for Changing the Configuration of the Watchdog Timer” on page 43.

In safety level 1, WDE is overridden by WDRF in MCUSR. See “MCUSR – MCU Status Register” on page 45 for description of WDRF. This means that WDE is always set when WDRF is set. To clear WDE, WDRF must be cleared before disabling the Watchdog with the procedure described above. This feature ensures multiple resets during conditions causing failure, and a safe start-up after the failure.

Note: If the watchdog timer is not going to be used in the application, it is important to go through a watchdog disable procedure in the initialization of the device. If the Watchdog is accidentally enabled, for example by a runaway pointer or brown-out condition, the device will be reset, which in turn will lead to a new watchdog reset. To avoid this situation, the application software should always clear the WDRF flag and the WDE control bit in the initialization routine.

- **Bits 5, 2..0 – WDP3..0: Watchdog Timer Prescaler 3, 2, 1, and 0**

The WDP3..0 bits determine the Watchdog Timer prescaling when the Watchdog Timer is enabled. The different prescaling values and their corresponding Timeout Periods are shown in Table 8-3 on page 47.

Table 8-3. Watchdog Timer Prescale Select

WDP3	WDP2	WDP1	WDP0	Number of WDT Oscillator Cycles	Typical Time-out at V_{CC} = 5.0V
0	0	0	0	2K cycles	16 ms
0	0	0	1	4K cycles	32 ms
0	0	1	0	8K cycles	64 ms
0	0	1	1	16K cycles	0.125 s
0	1	0	0	32K cycles	0.25 s
0	1	0	1	64K cycles	0.5 s
0	1	1	0	128K cycles	1.0 s
0	1	1	1	256K cycles	2.0 s
1	0	0	0	512K cycles	4.0 s
1	0	0	1	1024K cycles	8.0 s
1	0	1	0	Reserved[1]	
1	0	1	1		
1	1	0	0		
1	1	0	1		
1	1	1	0		
1	1	1	1		

Note: 1. If selected, one of the valid settings below 0b1010 will be used.

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).

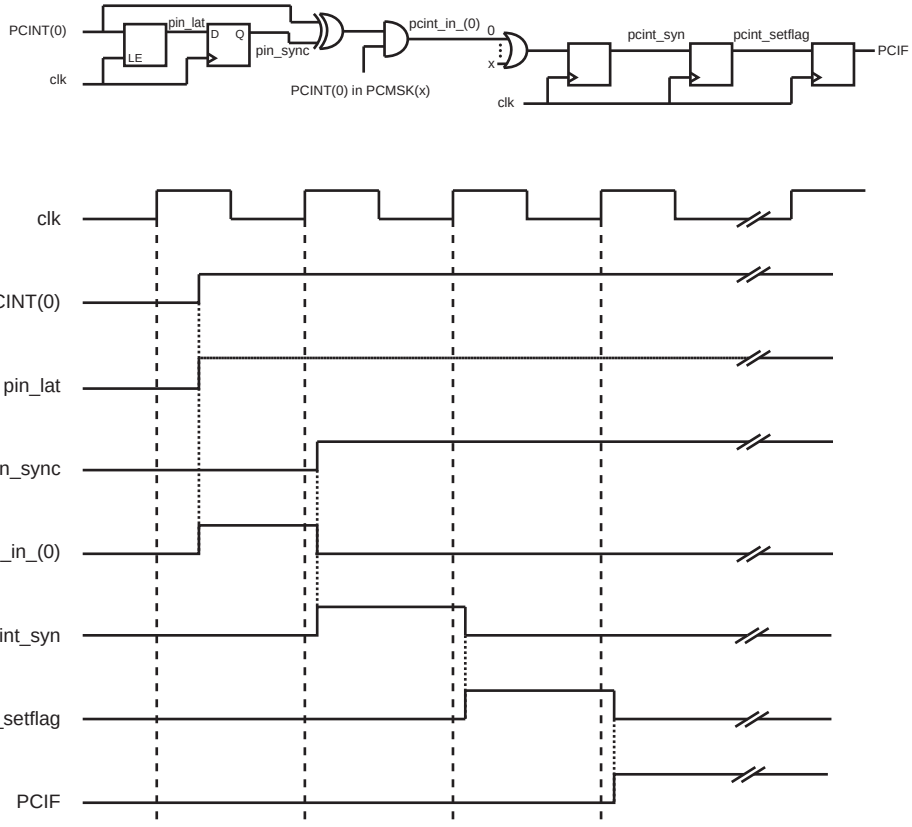
Note that if a level triggered interrupt is used for wake-up from Power-down, the required level must be held long enough for the MCU to complete the wake-up to trigger the level interrupt. If the level disappears before the end of the Start-up Time, the MCU will still wake up, but no interrupt will 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.

11.9 Register Description

11.9.1 TCCR0A – Timer/Counter Control Register A

Bit	7	6	5	4	3	2	1	0	
0x30 (0x50)	COM0A1	COM0A0	COM0B1	COM0B0	–	–	WGM01	WGM00	TCCR0A
Read/Write	R/W	R/W	R/W	R/W	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bits 7:6 – COM0A1:0: Compare Match Output A Mode**

These bits control the Output Compare pin (OC0A) behavior. If one or both of the COM0A1:0 bits are set, the OC0A output overrides the normal port functionality of the I/O pin it is connected to. However, note that the Data Direction Register (DDR) bit corresponding to the OC0A pin must be set in order to enable the output driver.

When OC0A is connected to the pin, the function of the COM0A1:0 bits depends on the WGM02:0 bit setting. Table 11-2 shows the COM0A1:0 bit functionality when the WGM02:0 bits are set to a normal or CTC mode (non-PWM).

Table 11-2. Compare Output Mode, non-PWM Mode

COM0A1	COM0A0	Description
0	0	Normal port operation, OC0A disconnected.
0	1	Toggle OC0A on Compare Match
1	0	Clear OC0A on Compare Match
1	1	Set OC0A on Compare Match

Table 11-3 shows the COM0A1:0 bit functionality when the WGM01:0 bits are set to fast PWM mode.

Table 11-3. Compare Output Mode, Fast PWM Mode[1]

COM0A1	COM0A0	Description
0	0	Normal port operation, OC0A disconnected.
0	1	WGM02 = 0: Normal Port Operation, OC0A Disconnected. WGM02 = 1: Toggle OC0A on Compare Match.
1	0	Clear OC0A on Compare Match, set OC0A at TOP
1	1	Set OC0A on Compare Match, clear OC0A at TOP

Note: 1. A special case occurs when OCR0A equals TOP and COM0A1 is set. In this case, the Compare Match is ignored, but the set or clear is done at TOP. See “Fast PWM Mode” on page 77 for more details.

Table 11-4 shows the COM0A1:0 bit functionality when the WGM02:0 bits are set to phase correct PWM mode.

Table 11-4. Compare Output Mode, Phase Correct PWM Mode(1)

COM0A1	COM0A0	Description
0	0	Normal port operation, OC0A disconnected.
0	1	WGM02 = 0: Normal Port Operation, OC0A Disconnected. WGM02 = 1: Toggle OC0A on Compare Match.
1	0	Clear OC0A on Compare Match when up-counting. Set OC0A on Compare Match when down-counting.
1	1	Set OC0A on Compare Match when up-counting. Clear OC0A on Compare Match when down-counting.

Note: 1. A special case occurs when OCR0A equals TOP and COM0A1 is set. In this case, the Compare Match is ignored, but the set or clear is done at TOP. See "Phase Correct PWM Mode" on page 79 for more details.

- **Bits 5:4 – COM0B1:0: Compare Match Output B Mode**

These bits control the Output Compare pin (OC0B) behavior. If one or both of the COM0B1:0 bits are set, the OC0B output overrides the normal port functionality of the I/O pin it is connected to. However, note that the Data Direction Register (DDR) bit corresponding to the OC0B pin must be set in order to enable the output driver.

When OC0B is connected to the pin, the function of the COM0B1:0 bits depends on the WGM02:0 bit setting. Table 11-5 shows the COM0B1:0 bit functionality when the WGM02:0 bits are set to a normal or CTC mode (non-PWM).

Table 11-5. Compare Output Mode, non-PWM Mode

COM0B1	COM0B0	Description
0	0	Normal port operation, OC0B disconnected.
0	1	Toggle OC0B on Compare Match
1	0	Clear OC0B on Compare Match
1	1	Set OC0B on Compare Match

Table 11-6 shows the COM0B1:0 bit functionality when the WGM02:0 bits are set to fast PWM mode.

Table 11-6. Compare Output Mode, Fast PWM Mode(1)

COM0B1	COM0B0	Description
0	0	Normal port operation, OC0B disconnected.
0	1	Reserved
1	0	Clear OC0B on Compare Match, set OC0B at TOP
1	1	Set OC0B on Compare Match, clear OC0B at TOP

Note: 1. A special case occurs when OCR0B equals TOP and COM0B1 is set. In this case, the Compare Match is ignored, but the set or clear is done at TOP. See "Fast PWM Mode" on page 77 for more details.

Table 11-7 shows the COM0B1:0 bit functionality when the WGM02:0 bits are set to phase correct PWM mode.

Table 11-7. Compare Output Mode, Phase Correct PWM Mode[1]

COM0B1	COM0B0	Description
0	0	Normal port operation, OCR0B disconnected.
0	1	Reserved
1	0	Clear ORC0B on Compare Match when up-counting. Set OCR0B on Compare Match when down-counting.
1	1	Set OCR0B on Compare Match when up-counting. Clear OCR0B on Compare Match when down-counting.

Note: 1. A special case occurs when OCR0B equals TOP and COM0B1 is set. In this case, the Compare Match is ignored, but the set or clear is done at TOP. See "Phase Correct PWM Mode" on page 79 for more details.

- **Bits 3, 2 – Res: Reserved Bits**

These bits are reserved bits in the ATtiny2313A/4313 and will always read as zero.

- **Bits 1:0 – WGM01:0: Waveform Generation Mode**

Combined with the WGM02 bit found in the TCCR0B Register, these bits control the counting sequence of the counter, the source for maximum (TOP) counter value, and what type of waveform generation to be used, see Table 11-8. Modes of operation supported by the Timer/Counter unit are: Normal mode (counter), Clear Timer on Compare Match (CTC) mode, and two types of Pulse Width Modulation (PWM) modes (see "Modes of Operation" on page 76).

Table 11-8. Waveform Generation Mode Bit Description

Mode	WGM2	WGM1	WGM0	Timer/Counter Mode of Operation	TOP	Update of OCRx at	TOV Flag Set on[1][2]
0	0	0	0	Normal	0xFF	Immediate	MAX
1	0	0	1	PWM, Phase Correct	0xFF	TOP	BOTTOM
2	0	1	0	CTC	OCR0A	Immediate	MAX
3	0	1	1	Fast PWM	0xFF	TOP	MAX
4	1	0	0	Reserved	–	–	–
5	1	0	1	PWM, Phase Correct	OCR0A	TOP	BOTTOM
6	1	1	0	Reserved	–	–	–
7	1	1	1	Fast PWM	OCR0A	TOP	TOP

Notes: 1. MAX = 0xFF
2. BOTTOM = 0x00

11.9.2 TCCR0B – Timer/Counter Control Register B

Bit	7	6	5	4	3	2	1	0	
0x33 (0x53)	FOC0A	FOC0B	–	–	WGM02	CS02	CS01	CS00	TCCR0B
Read/Write	W	W	R	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7 – FOC0A: Force Output Compare A**

The FOC0A bit is only active when the WGM bits specify a non-PWM mode.

However, for ensuring compatibility with future devices, this bit must be set to zero when TCCR0B is written when operating in PWM mode. When writing a logical one to the FOC0A bit, an immediate Compare Match is forced on the Waveform Generation unit. The OC0A output is changed according to its COM0A1:0 bits setting. Note that the FOC0A bit is implemented as a strobe. Therefore it is the value present in the COM0A1:0 bits that determines the effect of the forced compare.

A FOC0A strobe will not generate any interrupt, nor will it clear the timer in CTC mode using OCR0A as TOP.

The FOC0A bit is always read as zero.

- **Bit 6 – FOC0B: Force Output Compare B**

The FOC0B bit is only active when the WGM bits specify a non-PWM mode.

However, for ensuring compatibility with future devices, this bit must be set to zero when TCCR0B is written when operating in PWM mode. When writing a logical one to the FOC0B bit, an immediate Compare Match is forced on the Waveform Generation unit. The OC0B output is changed according to its COM0B1:0 bits setting. Note that the FOC0B bit is implemented as a strobe. Therefore it is the value present in the COM0B1:0 bits that determines the effect of the forced compare.

A FOC0B strobe will not generate any interrupt, nor will it clear the timer in CTC mode using OCR0B as TOP.

The FOC0B bit is always read as zero.

- **Bits 5:4 – Res: Reserved Bits**

These bits are reserved bits in the ATtiny2313A/4313 and will always read as zero.

- **Bit 3 – WGM02: Waveform Generation Mode**

See the description in the “TCCR0A – Timer/Counter Control Register A” on page 82.

- **Bits 2:0 – CS02:0: Clock Select**

The three Clock Select bits select the clock source to be used by the Timer/Counter. See Table 11-9 on page 86.

Table 11-9. Clock Select Bit Description

CS02	CS01	CS00	Description
0	0	0	No clock source (Timer/Counter stopped)
0	0	1	$clk_{I/O}$/(No prescaling)
0	1	0	$clk_{I/O}$/8 (From prescaler)
0	1	1	$clk_{I/O}$/64 (From prescaler)
1	0	0	$clk_{I/O}$/256 (From prescaler)
1	0	1	$clk_{I/O}$/1024 (From prescaler)
1	1	0	External clock source on T0 pin. Clock on falling edge.
1	1	1	External clock source on T0 pin. Clock on rising edge.

If external pin modes are used for the Timer/Counter0, transitions on the T0 pin will clock the counter even if the pin is configured as an output. This feature allows software control of the counting.

11.9.3 TCNT0 – Timer/Counter Register

Bit	7	6	5	4	3	2	1	0	
0x32 (0x52)	TCNT0[7:0]								TCNT0
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	

The Timer/Counter Register gives direct access, both for read and write operations, to the Timer/Counter unit 8-bit counter. Writing to the TCNT0 Register blocks (removes) the Compare Match on the following timer clock. Modifying the counter (TCNT0) while the counter is running, introduces a risk of missing a Compare Match between TCNT0 and the OCR0x Registers.

11.9.4 OCR0A – Output Compare Register A

Bit	7	6	5	4	3	2	1	0	
0x36 (0x56)	OCR0A[7:0]								OCR0A
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	

The Output Compare Register A contains an 8-bit value that is continuously compared with the counter value (TCNT0). A match can be used to generate an Output Compare interrupt, or to generate a waveform output on the OC0A pin.

11.9.5 OCR0B – Output Compare Register B

Bit	7	6	5	4	3	2	1	0	
0x3C (0x5C)	OCR0B[7:0]								OCR0B
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	

The Output Compare Register B contains an 8-bit value that is continuously compared with the counter value (TCNT0). A match can be used to generate an Output Compare interrupt, or to generate a waveform output on the OC0B pin.

11.9.6 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	

- **Bit 4 – Res: Reserved Bit**

This bit is reserved bit in the ATtiny2313A/4313 and will always read as zero.

- **Bit 2 – OCIE0B: Timer/Counter0 Output Compare Match B Interrupt Enable**

When the OCIE0B bit is written to one, and the I-bit in the Status Register is set, the Timer/Counter Compare Match B interrupt is enabled. The corresponding interrupt is executed if a Compare Match in Timer/Counter occurs, i.e., when the OCF0B bit is set in the Timer/Counter Interrupt Flag Register – TIFR.

- **Bit 1 – TOIE0: Timer/Counter0 Overflow Interrupt Enable**

When the TOIE0 bit is written to one, and the I-bit in the Status Register is set, the Timer/Counter0 Overflow interrupt is enabled. The corresponding interrupt is executed if an overflow in Timer/Counter0 occurs, i.e., when the TOV0 bit is set in the Timer/Counter 0 Interrupt Flag Register – TIFR.

- **Bit 0 – OCIE0A: Timer/Counter0 Output Compare Match A Interrupt Enable**

When the OCIE0A bit is written to one, and the I-bit in the Status Register is set, the Timer/Counter0 Compare Match A interrupt is enabled. The corresponding interrupt is executed if a Compare Match in Timer/Counter0 occurs, i.e., when the OCF0A bit is set in the Timer/Counter 0 Interrupt Flag Register – TIFR.

11.9.7 TIFR – Timer/Counter Interrupt Flag Register

Bit	7	6	5	4	3	2	1	0	
0x38 (0x58)	TOV1	OCF1A	OCF1B	–	ICF1	OCF0B	TOV0	OCF0A	TIFR
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	

- **Bit 4 – Res: Reserved Bit**

This bit is reserved bit in the ATtiny2313A/4313 and will always read as zero.

- **Bit 2 – OCF0B: Output Compare Flag 0 B**

The OCF0B bit is set when a Compare Match occurs between the Timer/Counter and the data in OCR0B – Output Compare Register0 B. OCF0B is cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, OCF0B is cleared by writing a logic one to the flag. When the I-bit in SREG, OCIE0B (Timer/Counter Compare B Match Interrupt Enable), and OCF0B are set, the Timer/Counter Compare Match Interrupt is executed.

- **Bit 1 – TOV0: Timer/Counter0 Overflow Flag**

The bit TOV0 is set when an overflow occurs in Timer/Counter0. TOV0 is cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, TOV0 is cleared by writing a logic one to the flag. When the SREG I-bit, TOIE0 (Timer/Counter0 Overflow Interrupt Enable), and TOV0 are set, the Timer/Counter0 Overflow interrupt is executed.

The setting of this flag is dependent of the WGM02:0 bit setting. Refer to Table 11-8, “Waveform Generation Mode Bit Description” on page 84.

- **Bit 0 – OCF0A: Output Compare Flag 0 A**

The OCF0A bit is set when a Compare Match occurs between the Timer/Counter0 and the data in OCR0A – Output Compare Register0 A. OCF0A is cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, OCF0A is cleared by writing a logic one to the flag. When the I-bit in SREG, OCIE0A (Timer/Counter0 Compare Match Interrupt Enable), and OCF0A are set, the Timer/Counter0 Compare Match Interrupt is executed.