Universidad de Costa Rica
Escuela de Ingeniería Eléctrica
IE0624 - Laboratorio de Microcontroladores

Laboratorio 2

Sebastián Vargas Quesada, C18295

18 de septiembre del 2024, Ciclo II

1. Resumen

En el presente laboratorio se desarrolló un juego de memoria utilizando el microcontrolador ATtiny4313. Para su implementación, se emplearon componentes electrónicos adicionales, como resistencias y LEDs de diferentes colores. Además, se utilizaron timers e interrupciones para optimizar la eficiencia del código. La lógica del programa fue codificada en lenguaje C, implementada mediante una máquina de estados, y los resultados se simularon en Simulide. Durante el desarrollo, la implementación de las interrupciones en los botones representó un desafío significativo. Sin embargo, se logró implementar satisfactoriamente la funcionalidad del código. El repositorio del proyecto se puede consultar en el siguiente enlace: https://github.com/sebasvq106/microcontroladores

2. Nota Teorica

2.1. Información General del Microcontrolador

Para la realización de este laboratorio se utilizará el microcontrolador ATtiny4313, un dispositivo de 8 bits de la familia AVR, fabricado por Microchip Technology. Este microcontrolador se basa en la arquitectura RISC y es conocido por su bajo consumo de energía y su versatilidad en aplicaciones de electrónica. Aunque su capacidad de memoria es limitada en comparación con microcontroladores más avanzados, su costo y facilidad de uso lo hacen ideal para proyectos de menor complejidad.

En este laboratorio, nos enfocaremos en el uso de los pines de entrada/salida (GPIO), temporizadores y interrupciones que ofrece el ATtiny4313. Este microcontrolador cuenta con hasta 18 pines de GPIO, que se pueden configurar según las necesidades de la aplicación. Estos pines son capaces de funcionar como entradas digitales o salidas, y también pueden ser utilizados para funciones especiales como salidas PWM y comunicación serial.

2.2. Diagrama de Bloques

A continuación se muestra el diagrama de bloques del microcontrolador:

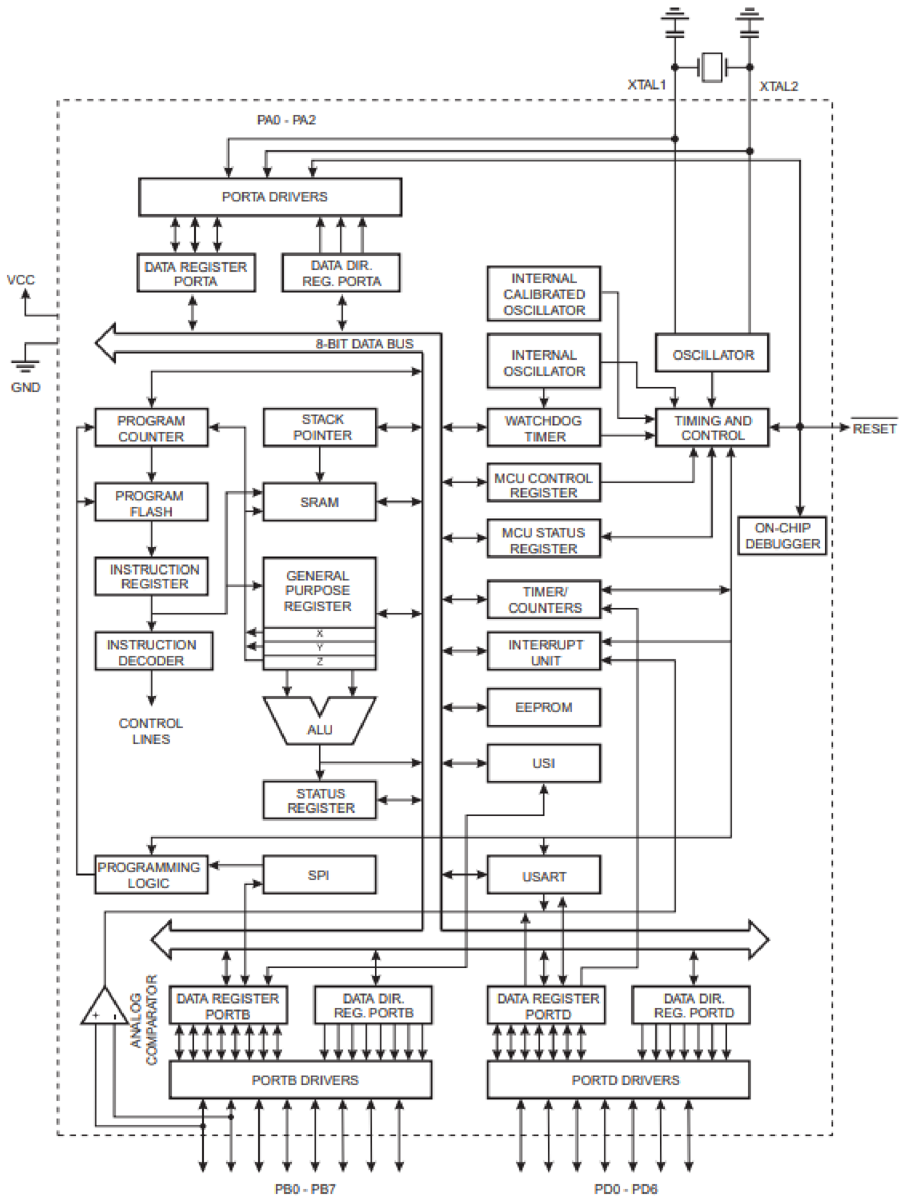


Figura 1: Diagrama de Bloques del ATtiny4313

2.3. Diagrama de Pines

A continuación se muestra el diagrama de pines del microcontrolador:

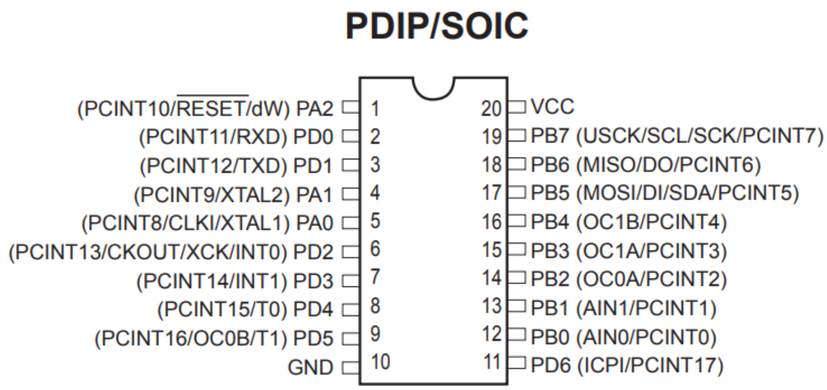


Figura 2: Diagrama de Pines del ATtiny4313

2.4. Periféricos utilizados

Para este laboratorio se utilizarán los diferentes pines de entrada y salida que posee el microcontrolador, sin embargo estos se tienen que configurar para que el circuito funcione, se utilizaron los pines B para las salidas y los pines D para las entradas, además, se tiene que configurar las resistencias pull-up internas. A continuación se muestra el código para confirgurar los diferentes pines:

```
// Configurar los pines PB3, PB4, PB5, y PB6 como salidas para los LEDs
 DDRB = (1 << PB3) | (1 << PB4) | (1 << PB5) | (1 << PB6);

 // Configurar los pines PD2, PD3, PD4 y PD5 como entradas para los
    botones
 DDRD &= ~((1 << PD2) | (1 << PD3) | (1 << PD4) | (1 << PD5));

 // Habilitar resistencias pull-up internas
 PORTD |= (1 << PD2) | (1 << PD3) | (1 << PD4) | (1 << PD5);
```

2.5. Componentes Electrónicos Complementarios

Para este laboratorio se utilizaron 4 diferentes LED's (Rojo, Verde, Amarrillo y Azul) los cuales muestran la secuencia. Por otra parte, se realizo un presupuesto para realizar el laboratorio, por lo que en la siguiente tabla se las los precios aproximados de los componentes a utilizar:

Componentes	**Precio aproximado**
ATtiny4313	1.5 USD
8 x resistores	2 USD
4 x capacitor	2 USD
4 x LED'S	1 USD

3. Desarrollo

3.1. Implemetación del botones y los Leds

Primero, se ensambló el circuito con todas sus conexiones y componentes. Luego, se implementó un pequeño código para verificar el funcionamiento de cada botón y LED. Este código enciende el LED correspondiente al color del botón que se presiona. A continuación, se muestra una imagen del circuito y el código utilizado.

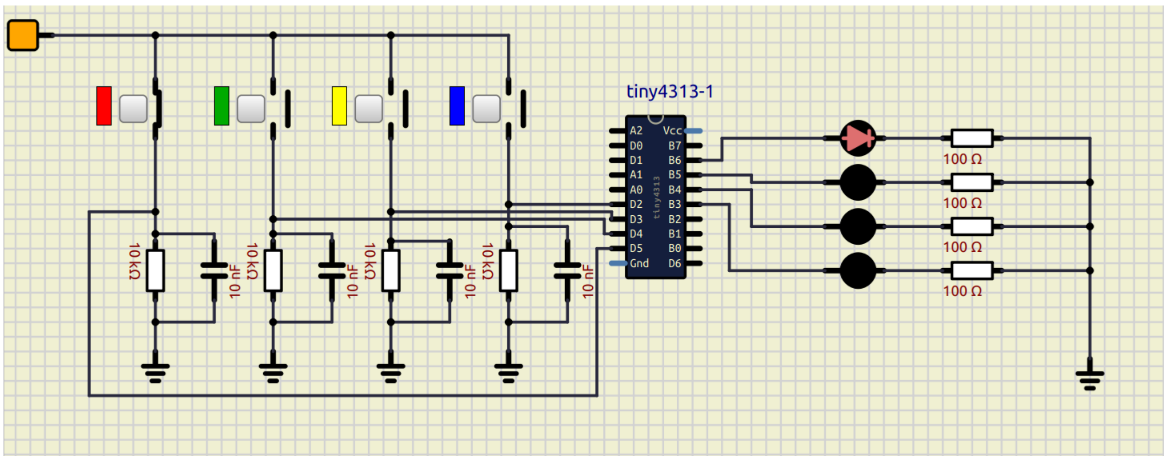


Figura 3: Implementación del botones y los leds

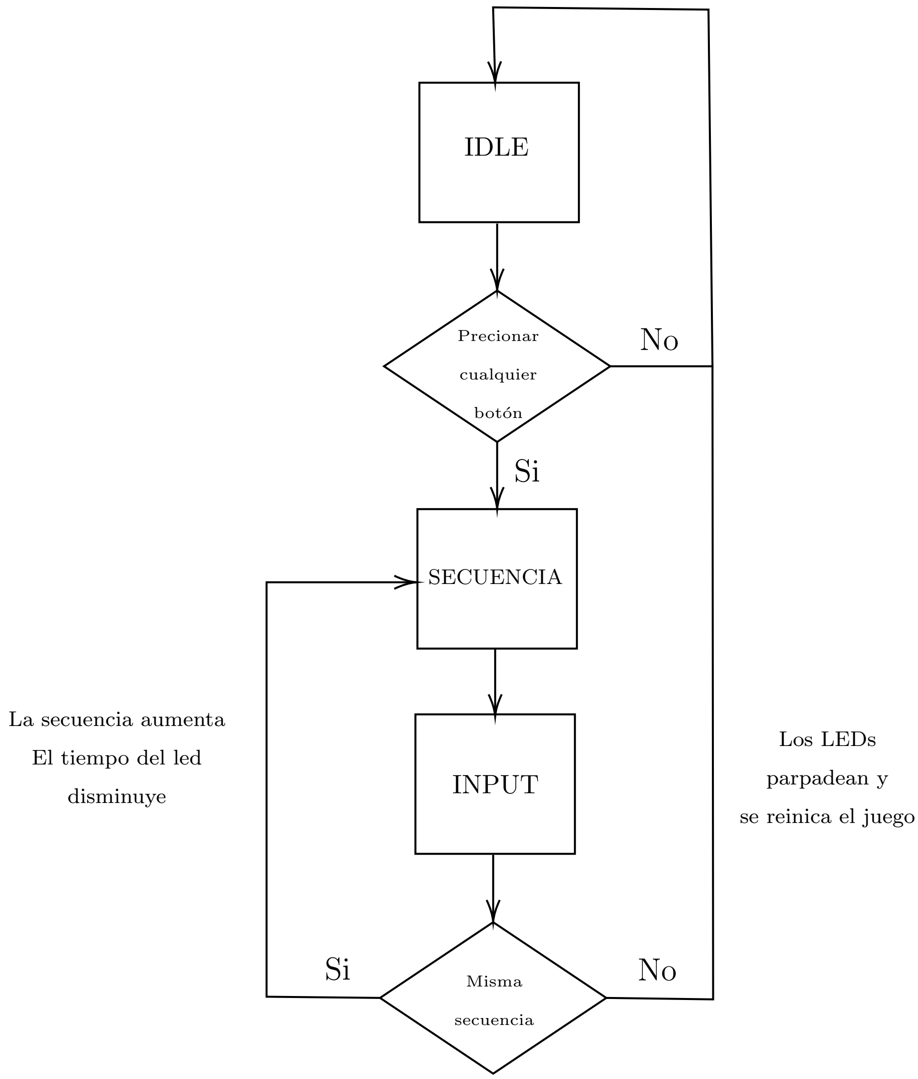
```c
#include <avr/io.h>
#include <util/delay.h>

int main(void)
{
    // Configurar los pines de salida
    DDRB = (1 << PB3) | (1 << PB4) | (1 << PB5) | (1 << PB6);

    // Configurar los pines entrada
    DDRD &= ~((1 << PD2) | (1 << PD3) | (1 << PD4) | (1 << PD5));

    // Activar resistencias pull-up para los botones
    PORTD |= (1 << PD2) | (1 << PD3) | (1 << PD4) | (1 << PD5);

    while (1) {

        // Boton 1 RED
        if (PIND & (1 << PD5)) {
            PORTB |= (1 << PB6);    // Encender LED
        } else {
            PORTB &= ~(1 << PB6);   // Apagar LED
        }

        // Boton 2 GREEN
```

```c
        if (PIND & (1 << PD4)) {
            PORTB |= (1 << PB5);
        } else {
            PORTB &= ~(1 << PB5);
        }

        // Boton 3 YELLOW
        if (PIND & (1 << PD3)) {
            PORTB |= (1 << PB4);
        } else {
            PORTB &= ~(1 << PB4);
        }

        // Boton 4 BLUE
        if (PIND & (1 << PD2)) {
            PORTB |= (1 << PB3);
        } else {
            PORTB &= ~(1 << PB3);
        }
    }
}
```

3.2. Implementación de la FSM, Timers e Interrupciones

3.2.1. Máquina de Estados

Para resolver el problema de este laboratorio, se definieron tres estados: IDLE, SECUENCIA e INPUT. El estado IDLE es el estado inicial de la máquina de estados, en el cual se espera a que se presione un botón para comenzar el parpadeo de los LED y luego pasar al estado SECUENCIA. El estado SECUENCIA se encarga de mostrar la secuencia de LED y, al finalizar, transita al estado INPUT. En el estado INPUT, se leen las entradas de los botones y se verifica si coinciden con la secuencia mostrada. Si la secuencia es correcta, se regresa al estado SECUENCIA, incrementando en 1 el tamaño de la secuencia y reduciendo el tiempo de visualización del LED. Si la secuencia es incorrecta, se muestra un parpadeo de error y se regresa al estado IDLE.



3.2.2. Timers

A diferencia del laboratorio 1, donde se utilizaron retardos con contadores, en este laboratorio se emplearán los timers integrados en el microcontrolador ATtiny4313, ya que ofrecen una manera más eficiente de generar retardos en el programa. En mi implementación, habrá tres tipos de retardos: uno para el parpadeo de los LED, un retardo dinámico para la secuencia, y un retardo después de presionar un botón. A continuación, se muestra una parte del código que configura el timer:

```
// Configurar el Timer1 en modo CTC
void setup_timer1() {
    TCCR1B |= (1 << WGM12); // Configurar Timer1 en modo CTC
    TCCR1B |= (1 << CS12) | (1 << CS10); // Prescaler de 1024
    OCR1A = 156; // Valor de comparacion en 10ms
    TIMSK |= (1 << OCIE1A);// Habilitar interrupcion por comparacion
}
```

En este código, el timer se configura en modo CTC (Clear Timer on Compare Match) para generar una interrupción cada 10 ms. Esto es necesario porque se manejan tres tipos de retardos diferentes a lo largo del programa.

3.2.3. Interrupciones

Para este laboratorio se implementan interrupciones tanto para los timers como para los botones en el microcontrolador ATtiny4313. Las interrupciones permiten gestionar los eventos de manera más eficiente, ya que el microcontrolador no necesita estar constantemente verificando el estado de los botones. En este caso, se configuran interrupciones para los timers, que controlan los retardos en el parpadeo de los LED y la secuencia dinámica, así como para los botones, detectando cuándo se presionan sin necesidad de utilizar recursos constantemente.

```
// ISR para manejar los contadores de los delays
ISR(TIMER1_COMPA_vect) {
    if (delay_counter_blink > 0) {
        delay_counter_blink--;
    }

    if (delay_counter_sequence > 0) {
        delay_counter_sequence--;
    }

    if (delay_counter_button > 0) {
        delay_counter_button--;
    }
}
```

En este código se puede observar la interrupción para el timer el cual verifica cual retardo se tiene que ejecutar.

3.3. Resultados y Análisis

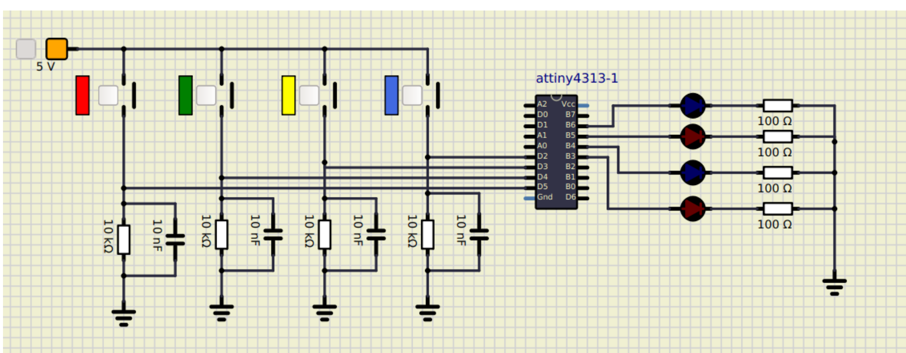


Figura 4: Circuito antes de accionar algún botón

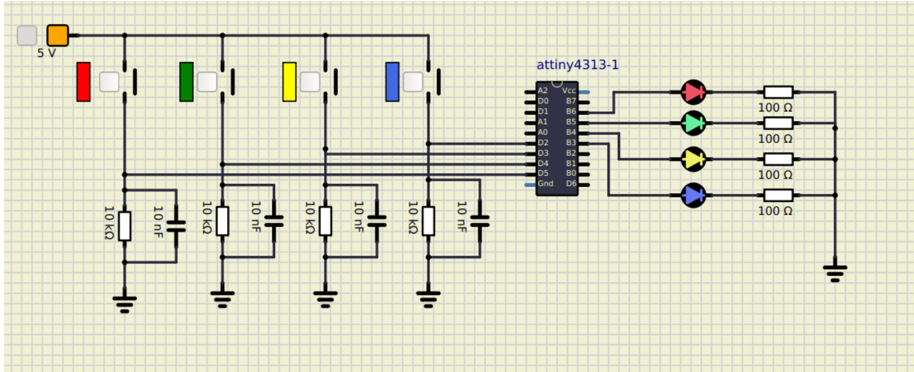


Figura 5: Circuito luego de accionar algún botón

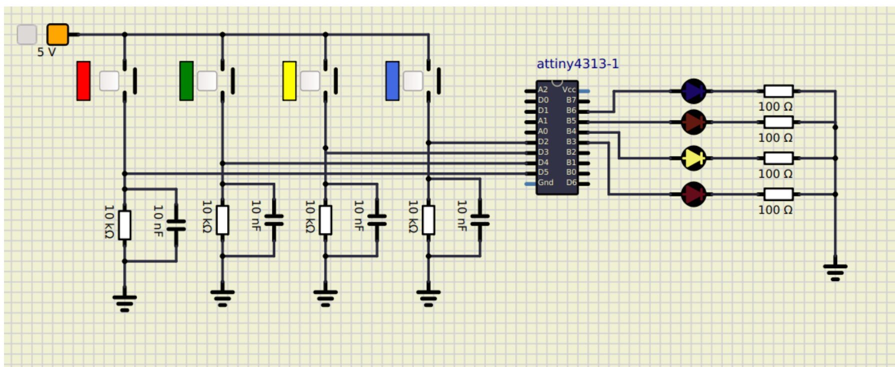


Figura 6: Circuito mostrando la secuencia

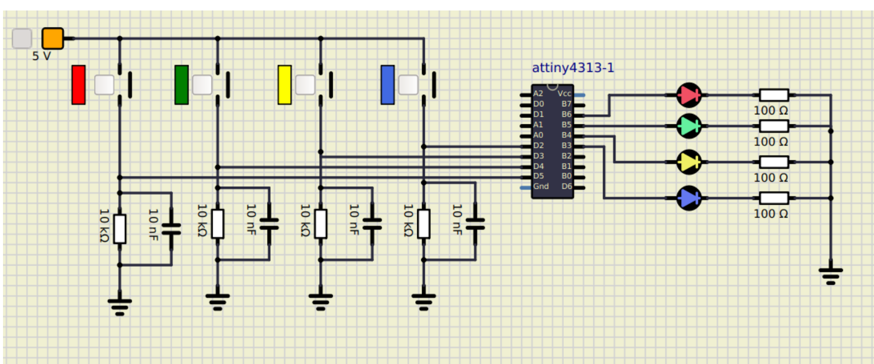


Figura 7: Circuito luego fallar la secuencia

Tras analizar los resultados obtenidos, se puede confirmar que el circuito funciona según lo esperado. Al presionar un botón, los LEDs se encenderán y apagarán dos veces, dando paso a la secuencia. Una vez finalizada la secuencia, se deberá repetir correctamente con los botones para avanzar de ronda, lo que aumentará la longitud de la secuencia en 1 y reducirá el tiempo de encendido en 200 ms. Si la secuencia ingresada es incorrecta, el circuito hará parpadear los LEDs tres veces como señal de error.

4. Conclusiones y Recomendaciones

- Se logró implementar una máquina de estados con tres estados, permitiendo abarcar de manera eficiente todos los casos planteados en el laboratorio. Cada estado fue diseñado para gestionar una situación específica, asegurando un comportamiento lógico y consistente del sistema.
- Se utilizaron timers para gestionar los retardos, lo que permitió mejorar la eficiencia del programa al evitar bloqueos en la ejecución y asegurar un control más preciso sobre los tiempos de respuesta.
- Se implementaron interrupciones para los timers con el objetivo de gestionar los tiempos de manera eficiente sin bloquear la ejecución del programa, permitiendo que el sistema responda de manera inmediata a otros eventos mientras se controlan los retardos de forma precisa.
- Se recomienda implementar primero la lógica de la máquina de estados y, posteriormente, integrar los timers y las interrupciones. Este enfoque permite estructurar el código de manera más organizada y reduce la probabilidad de errores, facilitando la depuración y el correcto funcionamiento del sistema.

Referencias

[1] Atmel. (2011). *ATtiny4313 Datasheet.*Recuperado el 08 de septiembre del 2024, de https://www.alldatasheet.com/view.jsp?Searchword=Attiny4313%20datasheet&gad_source=1&gclid=CjwKCAjw_4S3BhAAEiwA_64YhpnRxT-pNLr4atFQiJn-fhvba7MlaChRUPGU_wH4ctzj1Tp95sfdlRoC3UkQAvD_BwE.

5. Apéndices

Se presentán los datasheets de los componetes utilizados

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

Preliminary

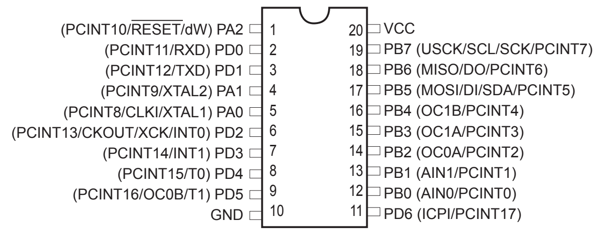
Summary

Rev. 8246AS–AVR–11/09

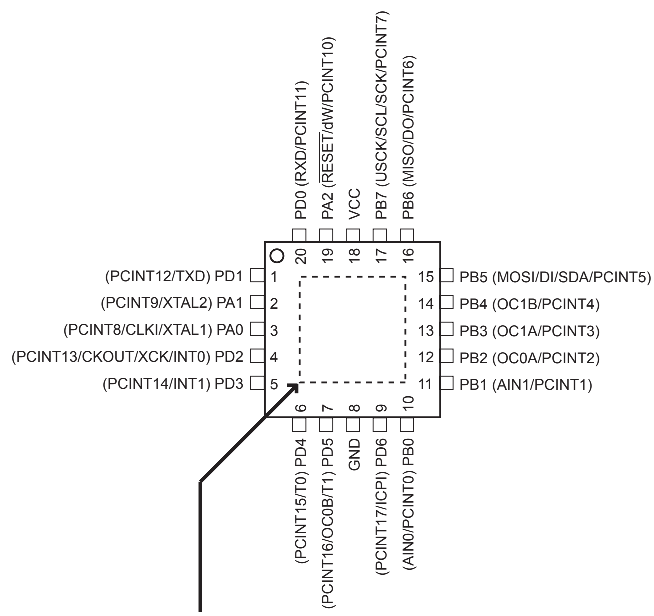
1. Pin Configurations

Figure 1-1. Pinout ATtiny2313A/4313

PDIP/SOIC



MLF/VQFN



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 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 61.

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 62.

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 66.

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 21-3 on page 198. 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

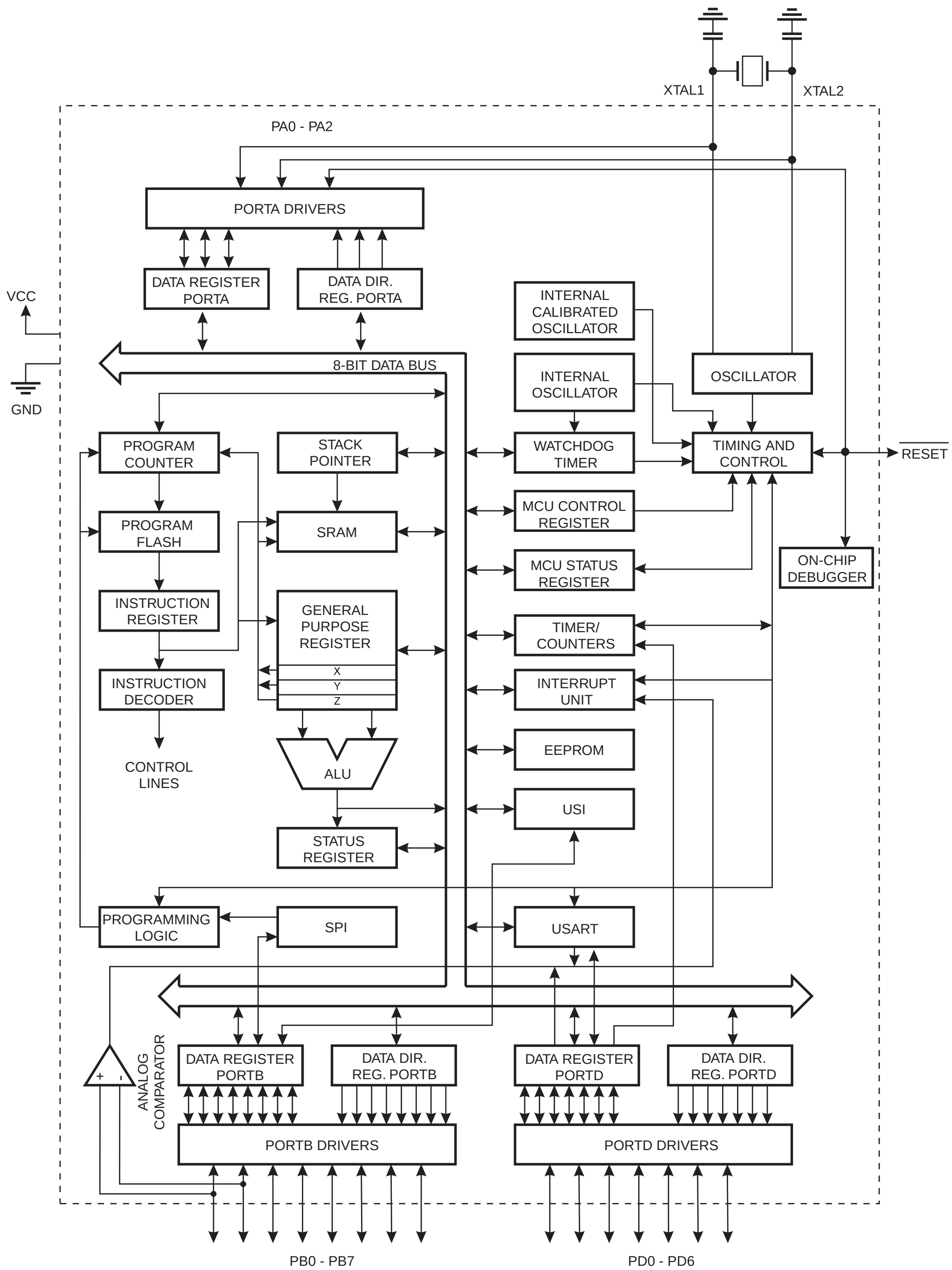
Output from the inverting Oscillator amplifier. XTAL2 is an alternate function for PA1.

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.

3.4 Disclaimer

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

4. Register Summary

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Page
0x3F (0x5F)	SREG	I	T	H	S	V	N	Z	C	8
0x3E (0x5E)	Reserved	–	–	–	–	–	–	–	–	
0x3D (0x5D)	SPL	SP7	SP6	SP5	SP4	SP3	SP2	SP1	SP0	11
0x3C (0x5C)	OCR0B	Timer/Counter0 – Compare Register B								85
0x3B (0x5B)	GIMSK	INT1	INT0	PCIE0	PCIE2	PCIE1	–	–	–	50
0x3A (0x5A)	GIFR	INTF1	INTF0	PCIF0	PCIF2	PCIF1	–	–	–	51
0x39 (0x59)	TIMSK	TOIE1	OCIE1A	OCIE1B	–	ICIE1	OCIE0B	TOIE0	OCIE0A	86, 115
0x38 (0x58)	TIFR	TOV1	OCF1A	OCF1B	–	ICF1	OCF0B	TOV0	OCF0A	86, 115
0x37 (0x57)	SPMCSR	–	–	RSIG	CTPB	RFLB	PGWRT	PGERS	SPMEN	176
0x36 (0x56)	OCR0A	Timer/Counter0 – Compare Register A								85
0x35 (0x55)	MCUCR	PUD	SM1	SE	SM0	ISC11	ISC10	ISC01	ISC00	36, 50, 68
0x34 (0x54)	MCUSR	–	–	–	–	WDRF	BORF	EXTRF	PORF	44
0x33 (0x53)	TCCR0B	FOC0A	FOC0B	–	–	WGM02	CS02	CS01	CS00	84
0x32 (0x52)	TCNT0	Timer/Counter0 (8-bit)								85
0x31 (0x51)	OSCCAL	–	CAL6	CAL5	CAL4	CAL3	CAL2	CAL1	CAL0	30
0x30 (0x50)	TCCR0A	COM0A1	COM0A0	COM0B1	COM0B0	–	–	WGM01	WGM00	81
0x2F (0x4F)	TCCR1A	COM1A1	COM1A0	COM1B1	COM1B0	–	–	WGM11	WGM10	110
0x2E (0x4E)	TCCR1B	ICNC1	ICES1	–	WGM13	WGM12	CS12	CS11	CS10	112
0x2D (0x4D)	TCNT1H	Timer/Counter1 – Counter Register High Byte								114
0x2C (0x4C)	TCNT1L	Timer/Counter1 – Counter Register Low Byte								114
0x2B (0x4B)	OCR1AH	Timer/Counter1 – Compare Register A High Byte								114
0x2A (0x4A)	OCR1AL	Timer/Counter1 – Compare Register A Low Byte								114
0x29 (0x49)	OCR1BH	Timer/Counter1 – Compare Register B High Byte								114
0x28 (0x48)	OCR1BL	Timer/Counter1 – Compare Register B Low Byte								114
0x27 (0x47)	Reserved	–	–	–	–	–	–	–	–	
0x26 (0x46)	CLKPR	CLKPCE	–	–	–	CLKPS3	CLKPS2	CLKPS1	CLKPS0	30
0x25 (0x45)	ICR1H	Timer/Counter1 - Input Capture Register High Byte								114
0x24 (0x44)	ICR1L	Timer/Counter1 - Input Capture Register Low Byte								114
0x23 (0x43)	GTCCR	–	–	–	–	–	–	–	PSR10	118
0x22 (ox42)	TCCR1C	FOC1A	FOC1B	–	–	–	–	–	–	113
0x21 (0x41)	WDTCSR	WDIF	WDIE	WDP3	WDCE	WDE	WDP2	WDP1	WDP0	44
0x20 (0x40)	PCMSK0	PCINT7	PCINT6	PCINT5	PCINT4	PCINT3	PCINT2	PCINT1	PCINT0	53
0x1F (0x3F)	Reserved	–	–	–	–	–	–	–	–	
0x1E (0x3E)	EEAR	–	EEPROM Address Register							22
0x1D (0x3D)	EEDR	EEPROM Data Register								22
0x1C (0x3C)	EECR	–	–	EEPM1	EEPM0	EERIE	EEMPE	EEPE	EERE	22
0x1B (0x3B)	PORTA	–	–	–	–	–	PORTA2	PORTA1	PORTA0	68
0x1A (0x3A)	DDRA	–	–	–	–	–	DDA2	DDA1	DDA0	68
0x19 (0x39)	PINA	–	–	–	–	–	PINA2	PINA1	PINA0	69
0x18 (0x38)	PORTB	PORTB7	PORTB6	PORTB5	PORTB4	PORTB3	PORTB2	PORTB1	PORTB0	69
0x17 (0x37)	DDRB	DDB7	DDB6	DDB5	DDB4	DDB3	DDB2	DDB1	DDB0	69
0x16 (0x36)	PINB	PINB7	PINB6	PINB5	PINB4	PINB3	PINB2	PINB1	PINB0	69
0x15 (0x35)	GPIOR2	General Purpose I/O Register 2								23
0x14 (0x34)	GPIOR1	General Purpose I/O Register 1								23
0x13 (0x33)	GPIOR0	General Purpose I/O Register 0								23
0x12 (0x32)	PORTD	–	PORTD6	PORTD5	PORTD4	PORTD3	PORTD2	PORTD1	PORTD0	69
0x11 (0x31)	DDRD	–	DDD6	DDD5	DDD4	DDD3	DDD2	DDD1	DDD0	69
0x10 (0x30)	PIND	–	PIND6	PIND5	PIND4	PIND3	PIND2	PIND1	PIND0	69
0x0F (0x2F)	USIDR	USI Data Register								165
0x0E (0x2E)	USISR	USISIF	USIOIF	USIPF	USIDC	USICNT3	USICNT2	USICNT1	USICNT0	164
0x0D (0x2D)	USICR	USISIE	USIOIE	USIWM1	USIWM0	USICS1	USICS0	USICLK	USITC	162
0x0C (0x2C)	UDR	UART Data Register (8-bit)								136
0x0B (0x2B)	UCSRA	RXC	TXC	UDRE	FE	DOR	UPE	U2X	MPCM	137
0x0A (0x2A)	UCSRB	RXCIE	TXCIE	UDRIE	RXEN	TXEN	UCSZ2	RXB8	TXB8	138
0x09 (0x29)	UBRRL	UBRRH[7:0]								140
0x08 (0x28)	ACSR	ACD	ACBG	ACO	ACI	ACIE	ACIC	ACIS1	ACIS0	167
0x07 (0x27)	BODCR	–	–	–	–	–	–	BODS	BODSE	37
0x06 (0x26)	PRR	–	–	–	–	PRTIM1	PRTIM0	PRUSI	PRUSART	36
0x05 (0x25)	PCMSK2	–	PCINT17	PCINT16	PCINT15	PCINT14	PCINT13	PCINT12	PCINT11	52
0x04 (0x24)	PCMSK1	–	–	–	–	–	PCINT10	PCINT9	PCINT8	52
0x03 (0x23)	UCSRC	UMSEL1	UMSEL0	UPM1	UPM0	USBS	UCSZ1	UCSZ0	UCPOL	139
0x02 (0x22)	UBRRH	–	–	–	–	UBRRH[11:8]				140
0x01 (0x21)	DIDR	–	–	–	–	–	–	AIN1D	AIN0D	168
0x00 (0x20)	USIBR	USI Buffer Register								166

Notes: 1. For compatibility with future devices, reserved bits should be written to zero if accessed. Reserved I/O memory addresses should never be written.
2. I/O Registers within the address range 0x00 - 0x1F are directly bit-accessible using the SBI and CBI instructions. In these registers, the value of single bits can be checked by using the SBIS and SBIC instructions.
3. Some of the status flags are cleared by writing a logical one to them. Note that, unlike most other AVRs, the CBI and SBI instructions will only operate on the specified bit, and can therefore be used on registers containing such status flags. The CBI and SBI instructions work with registers 0x00 to 0x1F only.
4. When using the I/O specific commands IN and OUT, the I/O addresses 0x00 - 0x3F must be used. When addressing I/O Registers as data space using LD and ST instructions, 0x20 must be added to these addresses.

5. Instruction Set Summary

Mnemonics	Operands	Description	Operation	Flags	#Clocks
ARITHMETIC AND LOGIC INSTRUCTIONS					
ADD	Rd, Rr	Add two Registers	Rd ← Rd + Rr	Z,C,N,V,H	1
ADC	Rd, Rr	Add with Carry two Registers	Rd ← Rd + Rr + C	Z,C,N,V,H	1
ADIW	Rdl,K	Add Immediate to Word	Rdh:Rdl ← Rdh:Rdl + K	Z,C,N,V,S	2
SUB	Rd, Rr	Subtract two Registers	Rd ← Rd - Rr	Z,C,N,V,H	1
SUBI	Rd, K	Subtract Constant from Register	Rd ← Rd - K	Z,C,N,V,H	1
SBC	Rd, Rr	Subtract with Carry two Registers	Rd ← Rd - Rr - C	Z,C,N,V,H	1
SBCI	Rd, K	Subtract with Carry Constant from Reg.	Rd ← Rd - K - C	Z,C,N,V,H	1
SBIW	Rdl,K	Subtract Immediate from Word	Rdh:Rdl ← Rdh:Rdl - K	Z,C,N,V,S	2
AND	Rd, Rr	Logical AND Registers	Rd ← Rd • Rr	Z,N,V	1
ANDI	Rd, K	Logical AND Register and Constant	Rd ← Rd • K	Z,N,V	1
OR	Rd, Rr	Logical OR Registers	Rd ← Rd v Rr	Z,N,V	1
ORI	Rd, K	Logical OR Register and Constant	Rd ← Rd v K	Z,N,V	1
EOR	Rd, Rr	Exclusive OR Registers	Rd ← Rd ⊕ Rr	Z,N,V	1
COM	Rd	One's Complement	Rd ← 0xFF − Rd	Z,C,N,V	1
NEG	Rd	Two's Complement	Rd ← 0x00 − Rd	Z,C,N,V,H	1
SBR	Rd,K	Set Bit(s) in Register	Rd ← Rd v K	Z,N,V	1
CBR	Rd,K	Clear Bit(s) in Register	Rd ← Rd • (0xFF - K)	Z,N,V	1
INC	Rd	Increment	Rd ← Rd + 1	Z,N,V	1
DEC	Rd	Decrement	Rd ← Rd − 1	Z,N,V	1
TST	Rd	Test for Zero or Minus	Rd ← Rd • Rd	Z,N,V	1
CLR	Rd	Clear Register	Rd ← Rd ⊕ Rd	Z,N,V	1
SER	Rd	Set Register	Rd ← 0xFF	None	1
BRANCH INSTRUCTIONS					
RJMP	k	Relative Jump	PC ← PC + k + 1	None	2
IJMP		Indirect Jump to (Z)	PC ← Z	None	2
RCALL	k	Relative Subroutine Call	PC ← PC + k + 1	None	3
ICALL		Indirect Call to (Z)	PC ← Z	None	3
RET		Subroutine Return	PC ← STACK	None	4
RETI		Interrupt Return	PC ← STACK	I	4
CPSE	Rd,Rr	Compare, Skip if Equal	if (Rd = Rr) PC ← PC + 2 or 3	None	1/2/3
CP	Rd,Rr	Compare	Rd − Rr	Z, N,V,C,H	1
CPC	Rd,Rr	Compare with Carry	Rd − Rr − C	Z, N,V,C,H	1
CPI	Rd,K	Compare Register with Immediate	Rd − K	Z, N,V,C,H	1
SBRC	Rr, b	Skip if Bit in Register Cleared	if (Rr(b)=0) PC ← PC + 2 or 3	None	1/2/3
SBRS	Rr, b	Skip if Bit in Register is Set	if (Rr(b)=1) PC ← PC + 2 or 3	None	1/2/3
SBIC	P, b	Skip if Bit in I/O Register Cleared	if (P(b)=0) PC ← PC + 2 or 3	None	1/2/3
SBIS	P, b	Skip if Bit in I/O Register is Set	if (P(b)=1) PC ← PC + 2 or 3	None	1/2/3
BRBS	s, k	Branch if Status Flag Set	if (SREG(s) = 1) then PC←PC+k + 1	None	1/2
BRBC	s, k	Branch if Status Flag Cleared	if (SREG(s) = 0) then PC←PC+k + 1	None	1/2
BREQ	k	Branch if Equal	if (Z = 1) then PC ← PC + k + 1	None	1/2
BRNE	k	Branch if Not Equal	if (Z = 0) then PC ← PC + k + 1	None	1/2
BRCS	k	Branch if Carry Set	if (C = 1) then PC ← PC + k + 1	None	1/2
BRCC	k	Branch if Carry Cleared	if (C = 0) then PC ← PC + k + 1	None	1/2
BRSH	k	Branch if Same or Higher	if (C = 0) then PC ← PC + k + 1	None	1/2
BRLO	k	Branch if Lower	if (C = 1) then PC ← PC + k + 1	None	1/2
BRMI	k	Branch if Minus	if (N = 1) then PC ← PC + k + 1	None	1/2
BRPL	k	Branch if Plus	if (N = 0) then PC ← PC + k + 1	None	1/2
BRGE	k	Branch if Greater or Equal, Signed	if (N ⊕ V= 0) then PC ← PC + k + 1	None	1/2
BRLT	k	Branch if Less Than Zero, Signed	if (N ⊕ V= 1) then PC ← PC + k + 1	None	1/2
BRHS	k	Branch if Half Carry Flag Set	if (H = 1) then PC ← PC + k + 1	None	1/2
BRHC	k	Branch if Half Carry Flag Cleared	if (H = 0) then PC ← PC + k + 1	None	1/2
BRTS	k	Branch if T Flag Set	if (T = 1) then PC ← PC + k + 1	None	1/2
BRTC	k	Branch if T Flag Cleared	if (T = 0) then PC ← PC + k + 1	None	1/2
BRVS	k	Branch if Overflow Flag is Set	if (V = 1) then PC ← PC + k + 1	None	1/2
BRVC	k	Branch if Overflow Flag is Cleared	if (V = 0) then PC ← PC + k + 1	None	1/2
BRIE	k	Branch if Interrupt Enabled	if (I = 1) then PC ← PC + k + 1	None	1/2
BRID	k	Branch if Interrupt Disabled	if (I = 0) then PC ← PC + k + 1	None	1/2
BIT AND BIT-TEST INSTRUCTIONS					
SBI	P,b	Set Bit in I/O Register	I/O(P,b) ← 1	None	2
CBI	P,b	Clear Bit in I/O Register	I/O(P,b) ← 0	None	2
LSL	Rd	Logical Shift Left	Rd(n+1) ← Rd(n), Rd(0) ← 0	Z,C,N,V	1
LSR	Rd	Logical Shift Right	Rd(n) ← Rd(n+1), Rd(7) ← 0	Z,C,N,V	1
ROL	Rd	Rotate Left Through Carry	Rd(0)←C,Rd(n+1)← Rd(n),C←Rd(7)	Z,C,N,V	1

Mnemonics	Operands	Description	Operation	Flags	#Clocks
ROR	Rd	Rotate Right Through Carry	Rd(7)←C,Rd(n)← Rd(n+1),C←Rd(0)	Z,C,N,V	1
ASR	Rd	Arithmetic Shift Right	Rd(n) ← Rd(n+1), n=0..6	Z,C,N,V	1
SWAP	Rd	Swap Nibbles	Rd(3..0)←Rd(7..4),Rd(7..4)←Rd(3..0)	None	1
BSET	s	Flag Set	SREG(s) ← 1	SREG(s)	1
BCLR	s	Flag Clear	SREG(s) ← 0	SREG(s)	1
BST	Rr, b	Bit Store from Register to T	T ← Rr(b)	T	1
BLD	Rd, b	Bit load from T to Register	Rd(b) ← T	None	1
SEC		Set Carry	C ← 1	C	1
CLC		Clear Carry	C ← 0	C	1
SEN		Set Negative Flag	N ← 1	N	1
CLN		Clear Negative Flag	N ← 0	N	1
SEZ		Set Zero Flag	Z ← 1	Z	1
CLZ		Clear Zero Flag	Z ← 0	Z	1
SEI		Global Interrupt Enable	I ← 1	I	1
CLI		Global Interrupt Disable	I ← 0	I	1
SES		Set Signed Test Flag	S ← 1	S	1
CLS		Clear Signed Test Flag	S ← 0	S	1
SEV		Set Twos Complement Overflow.	V ← 1	V	1
CLV		Clear Twos Complement Overflow	V ← 0	V	1
SET		Set T in SREG	T ← 1	T	1
CLT		Clear T in SREG	T ← 0	T	1
SEH		Set Half Carry Flag in SREG	H ← 1	H	1
CLH		Clear Half Carry Flag in SREG	H ← 0	H	1
DATA TRANSFER INSTRUCTIONS					
MOV	Rd, Rr	Move Between Registers	Rd ← Rr	None	1
MOVW	Rd, Rr	Copy Register Word	Rd+1:Rd ← Rr+1:Rr	None	1
LDI	Rd, K	Load Immediate	Rd ← K	None	1
LD	Rd, X	Load Indirect	Rd ← (X)	None	2
LD	Rd, X+	Load Indirect and Post-Inc.	Rd ← (X), X ← X + 1	None	2
LD	Rd, - X	Load Indirect and Pre-Dec.	X ← X - 1, Rd ← (X)	None	2
LD	Rd, Y	Load Indirect	Rd ← (Y)	None	2
LD	Rd, Y+	Load Indirect and Post-Inc.	Rd ← (Y), Y ← Y + 1	None	2
LD	Rd, - Y	Load Indirect and Pre-Dec.	Y ← Y - 1, Rd ← (Y)	None	2
LDD	Rd,Y+q	Load Indirect with Displacement	Rd ← (Y + q)	None	2
LD	Rd, Z	Load Indirect	Rd ← (Z)	None	2
LD	Rd, Z+	Load Indirect and Post-Inc.	Rd ← (Z), Z ← Z+1	None	2
LD	Rd, -Z	Load Indirect and Pre-Dec.	Z ← Z - 1, Rd ← (Z)	None	2
LDD	Rd, Z+q	Load Indirect with Displacement	Rd ← (Z + q)	None	2
LDS	Rd, k	Load Direct from SRAM	Rd ← (k)	None	2
ST	X, Rr	Store Indirect	(X) ← Rr	None	2
ST	X+, Rr	Store Indirect and Post-Inc.	(X) ← Rr, X ← X + 1	None	2
ST	- X, Rr	Store Indirect and Pre-Dec.	X ← X - 1, (X) ← Rr	None	2
ST	Y, Rr	Store Indirect	(Y) ← Rr	None	2
ST	Y+, Rr	Store Indirect and Post-Inc.	(Y) ← Rr, Y ← Y + 1	None	2
ST	- Y, Rr	Store Indirect and Pre-Dec.	Y ← Y - 1, (Y) ← Rr	None	2
STD	Y+q,Rr	Store Indirect with Displacement	(Y + q) ← Rr	None	2
ST	Z, Rr	Store Indirect	(Z) ← Rr	None	2
ST	Z+, Rr	Store Indirect and Post-Inc.	(Z) ← Rr, Z ← Z + 1	None	2
ST	-Z, Rr	Store Indirect and Pre-Dec.	Z ← Z - 1, (Z) ← Rr	None	2
STD	Z+q,Rr	Store Indirect with Displacement	(Z + q) ← Rr	None	2
STS	k, Rr	Store Direct to SRAM	(k) ← Rr	None	2
LPM		Load Program Memory	R0 ← (Z)	None	3
LPM	Rd, Z	Load Program Memory	Rd ← (Z)	None	3
LPM	Rd, Z+	Load Program Memory and Post-Inc	Rd ← (Z), Z ← Z+1	None	3
SPM		Store Program Memory	(Z) ← R1:R0	None	-
IN	Rd, P	In Port	Rd ← P	None	1
OUT	P, Rr	Out Port	P ← Rr	None	1
PUSH	Rr	Push Register on Stack	STACK ← Rr	None	2
POP	Rd	Pop Register from Stack	Rd ← STACK	None	2
MCU CONTROL INSTRUCTIONS					
NOP		No Operation		None	1
SLEEP		Sleep	(see specific descr. for Sleep function)	None	1
WDR		Watchdog Reset	(see specific descr. for WDR/timer)	None	1
BREAK		Break	For On-chip Debug Only	None	N/A

6. Ordering Information

6.1 ATtiny2313A

Speed (MHz)	Power Supply	Ordering Code[(2)]	Package[(1)]	Operation Range
20[(3)]	1.8 - 5.5V	ATtiny2313A-PU ATtiny2313A-SU ATtiny2313A-MU ATtiny2313A-MMH[(4)(5)]	20P3 20S 20M1 20M2	Industrial (-40°C to 85°C)

Notes:
1. This device can also be supplied in wafer form. Please contact your local Atmel sales office for detailed ordering information and minimum quantities.
2. Pb-free packaging, complies to the European Directive for Restriction of Hazardous Substances (RoHS directive). Also Halide free and fully Green.
3. For Speed vs. V_{CC}, see "Speed Grades" on page 196.
4. NiPdAu finish
5. Topside marking for ATtiny2313A:
   - 1st Line: T2313
   - 2nd Line: Axx
   - 3rd Line: xxx

Package Type	
20P3	20-lead, 0.300" Wide, Plastic Dual Inline Package (PDIP)
20S	20-lead, 0.300" Wide, Plastic Gull Wing Small Outline Package (SOIC)
20M1	20-pad, 4 x 4 x 0.8 mm Body, Quad Flat No-Lead / Micro Lead Frame Package (MLF)
20M2	20-pad, 3 x 3 x 0.85 mm Body, Very Thin Quad Flat No Lead Package (VQFN)

6.2 ATtiny4313

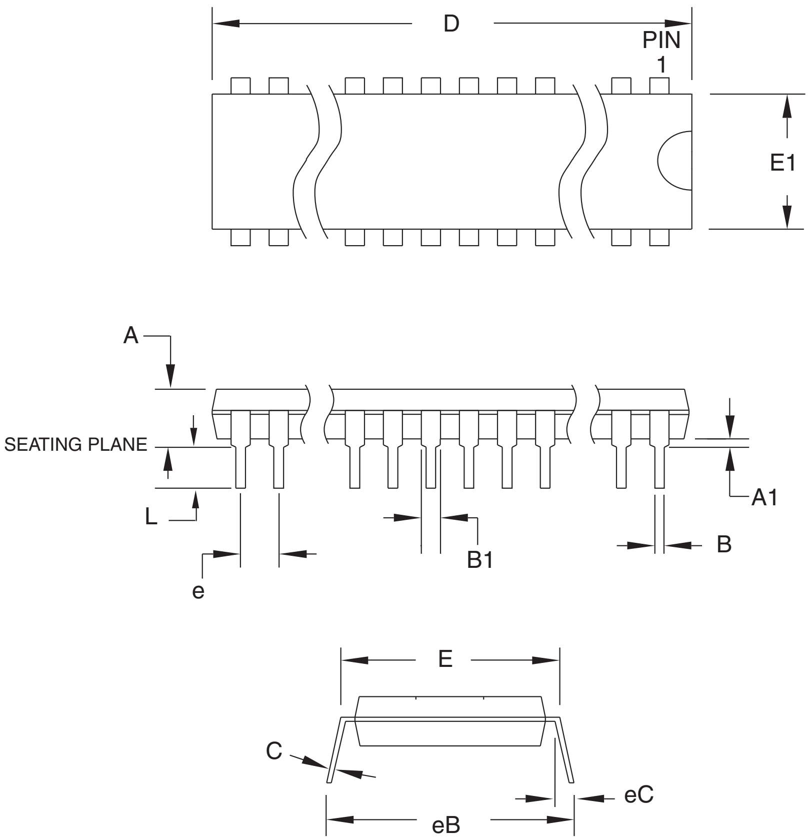
Speed (MHz)	Power Supply	Ordering Code[2]	Package[1]	Operation Range
20[3]	1.8 - 5.5V	ATtiny4313-PU ATtiny4313-SU ATtiny4313-MU ATtiny4313-MMH[4][5]	20P3 20S 20M1 20M2	Industrial (-40°C to 85°C)

Notes:
1. This device can also be supplied in wafer form. Please contact your local Atmel sales office for detailed ordering information and minimum quantities.
2. Pb-free packaging, complies to the European Directive for Restriction of Hazardous Substances (RoHS directive). Also Halide free and fully Green.
3. For Speed vs. V_{CC}, see "Speed Grades" on page 196.
4. NiPdAu finish
5. Topside marking for ATtiny4313:
   - 1st Line: T4313
   - 2nd Line: xx
   - 3rd Line: xxx

Package Type	
20P3	20-lead, 0.300" Wide, Plastic Dual Inline Package (PDIP)
20S	20-lead, 0.300" Wide, Plastic Gull Wing Small Outline Package (SOIC)
20M1	20-pad, 4 x 4 x 0.8 mm Body, Quad Flat No-Lead/Micro Lead Frame Package (MLF)
20M2	20-pad, 3 x 3 x 0.85 mm Body, Very Thin Quad Flat No Lead Package (VQFN)

7. Packaging Information

7.1 20P3



COMMON DIMENSIONS
(Unit of Measure = mm)

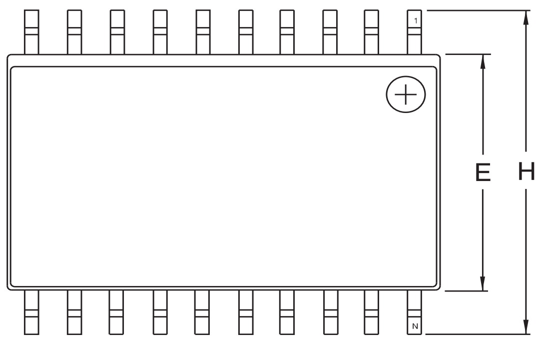
SYMBOL	MIN	NOM	MAX	NOTE
A	–	–	5.334	
A1	0.381	–	–	
D	25.493	–	25.984	Note 2
E	7.620	–	8.255	
E1	6.096	–	7.112	Note 2
B	0.356	–	0.559	
B1	1.270	–	1.551	
L	2.921	–	3.810	
C	0.203	–	0.356	
eB	–	–	10.922	
eC	0.000	–	1.524	
e	2.540 TYP			

Notes: 1. This package conforms to JEDEC reference MS-001, Variation AD.
2. Dimensions D and E1 do not include mold Flash or Protrusion. Mold Flash or Protrusion shall not exceed 0.25 mm (0.010").

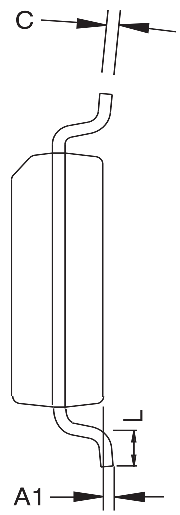
1/12/04

	TITLE	DRAWING NO.	REV.
2325 Orchard Parkway San Jose, CA 95131	**20P3**, 20-lead (0.300"/7.62 mm Wide) Plastic Dual Inline Package (PDIP)	20P3	C

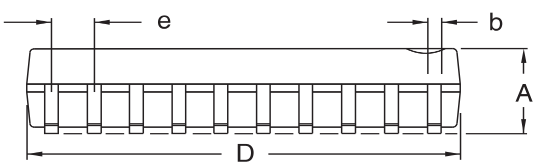
7.2 20S



Top View



End View



Side View

COMMON DIMENSIONS
(Unit of Measure = mm)

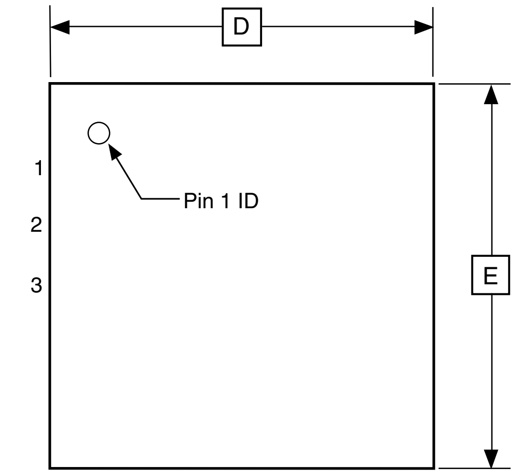
SYMBOL	MIN	NOM	MAX	NOTE
A	2.35		2.65	
A1	0.10		0.30	
b	0.33		0.51	4
C	0.23		0.32	
D	12.60		13.00	1
E	7.40		7.60	2
H	10.00		10.65	
L	0.40		1.27	3
e	1.27 BSC			

Notes.
1. This drawing is for general information only; refer to JEDEC Drawing MS-013, Variation AC for additional information.
2. Dimension 'D' does not include mold Flash, protrusions or gate burrs. Mold Flash, protrusions and gate burrs shall not exceed 0.15 mm (0.006') per side.
3. Dimension 'E' does not include inter-lead Flash or protrusion. Inter-lead Flash and protrusions shall not exceed 0.25 mm (0.010') per side.
4. 'L' is the length of the terminal for soldering to a substrate.
5. The lead width 'b', as measured 0.36 mm (0.014') or greater above the seating plane, shall not exceed a maximum value of 0.61 mm (0.024') per side.

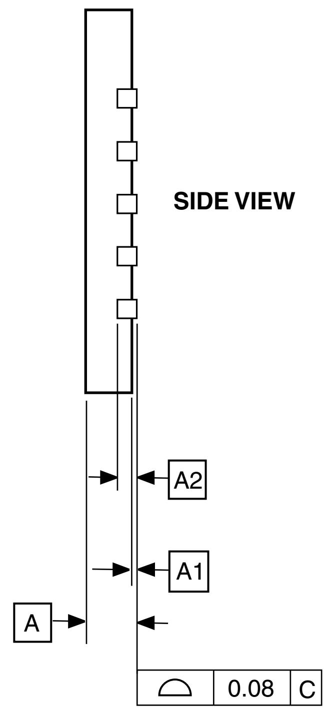
11/6/06

2325 Orchard Parkway San Jose, CA 95131	**TITLE** **20S2**, 20-lead, 0.300' Wide Body, Plastic Gull Wing Small Outline Package (SOIC)	**DRAWING NO.** 20S2	**REV.** B

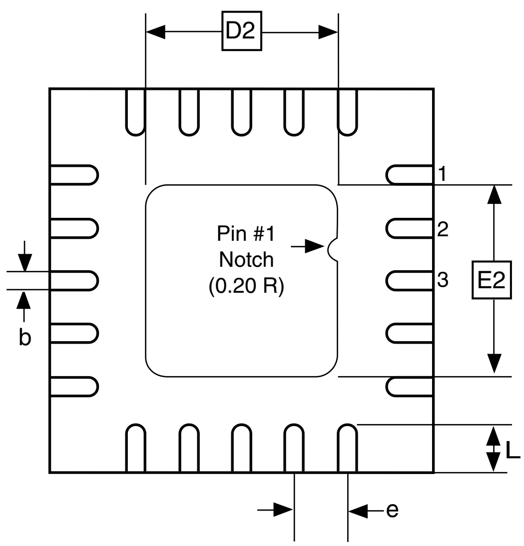
7.3 20M1



TOP VIEW



SIDE VIEW



BOTTOM VIEW

COMMON DIMENSIONS
(Unit of Measure = mm)

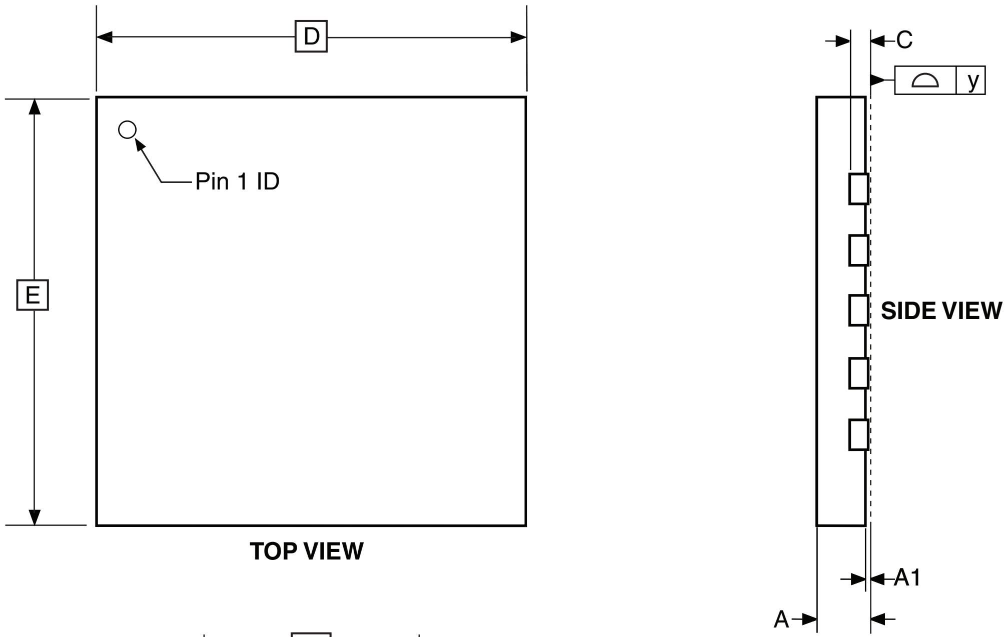
SYMBOL	MIN	NOM	MAX	NOTE
A	0.70	0.75	0.80	
A1	–	0.01	0.05	
A2		0.20 REF		
b	0.18	0.23	0.30	
D		4.00 BSC		
D2	2.45	2.60	2.75	
E		4.00 BSC		
E2	2.45	2.60	2.75	
e		0.50 BSC		
L	0.35	0.40	0.55	

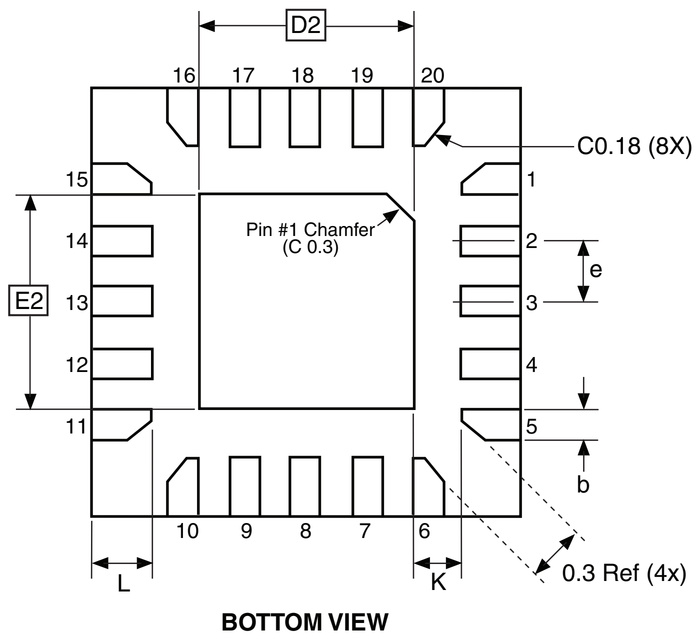
Note: Reference JEDEC Standard MO-220, Fig. 1 (SAW Singulation) WGGD-5.

10/27/04

	TITLE	DRAWING NO.	REV.
2325 Orchard Parkway San Jose, CA 95131	**20M1**, 20-pad, 4 x 4 x 0.8 mm Body, Lead Pitch 0.50 mm, 2.6 mm Exposed Pad, Micro Lead Frame Package (MLF)	20M1	A

7.4 20M2





COMMON DIMENSIONS
(Unit of Measure = mm)

SYMBOL	MIN	NOM	MAX	NOTE
A	0.75	0.80	0.85	
A1	0.00	0.02	0.05	
b	0.17	0.22	0.27	
C		0.152		
D	2.90	3.00	3.10	
D2	1.40	1.55	1.70	
E	2.90	3.00	3.10	
E2	1.40	1.55	1.70	
e	–	0.45	–	
L	0.35	0.40	0.45	
K	0.20	–	–	
y	0.00	–	0.08	

10/24/08

ATMEL Package Drawing Contact: packagedrawings@atmel.com	TITLE **20M2,** 20-pad, 3 x 3 x 0.85 mm Body, Lead Pitch 0.45 mm, 1.55 x 1.55 mm Exposed Pad, Thermally Enhanced Plastic Very Thin Quad Flat No Lead Package (VQFN)	GPC	DRAWING NO.	REV.
		ZFC	20M2	B

8. Errata

The revision letters in this section refer to the revision of the corresponding ATtiny2313A/4313 device.

8.1 ATtiny2313A

8.1.1 Rev. D

No known errata.

8.1.2 Rev. A – C

These device revisions were referred to as ATtiny2313/ATtiny2313V.

8.2 ATtiny4313

8.2.1 Rev. A

No known errata.

9. Datasheet Revision History

9.1 Rev. 8246A – 11/09

1. Initial revision. Created from document 2543_t2313.
2. Updated datasheet template.
3. Added VQFN in the Pinout Figure 1-1 on page 2.
4. Added Section 7.2 “Software BOD Disable” on page 34.
5. Added Section 7.3 “Power Reduction Register” on page 34.
6. Updated Table 7-2, “Sleep Mode Select,” on page 36.
7. Added Section 7.5.3 “BODCR – Brown-Out Detector Control Register” on page 37.
8. Added reset disable function in Figure 8-1 on page 38.
9. Added pin change interrupts PCINT1 and PCINT2 in Table 9-1 on page 47.
10. Added PCINT17..8 and PCMSK2..1 in Section 9.2 “External Interrupts” on page 48.
11. Added Section 9.3.4 “PCMSK2 – Pin Change Mask Register 2” on page 52.
12. Added Section 9.3.5 “PCMSK1 – Pin Change Mask Register 1” on page 52.
13. Updated Section 10.2.1 “Alternate Functions of Port A” on page 61.
14. Updated Section 10.2.2 “Alternate Functions of Port B” on page 62.
15. Updated Section 10.2.3 “Alternate Functions of Port D” on page 66.
16. Added UMSEL1 and UMSEL0 in Section 14.10.4 “UCSRC – USART Control and Status Register C” on page 139.
17. Added Section 15. “USART in SPI Mode” on page 145.
18. Added USI Buffer Register (USIBR) in Section 16.2 “Overview” on page 155 and in Figure 16-1 on page 155.
19. Added Section 16.5.4 “USIBR – USI Buffer Register” on page 166.
20. Updated Section 19.6.3 “Reading Device Signature Imprint Table from Firmware” on page 175.
21. Updated Section 19.9.1 “SPMCSR – Store Program Memory Control and Status Register” on page 176.
22. Added Section 20.3 “Device Signature Imprint Table” on page 180.
23. Updated Section 20.3.1 “Calibration Byte” on page 181.
24. Changed BS to BS1 in Section 20.6.13 “Reading the Signature Bytes” on page 189.
25. Updated Section 21.2 “DC Characteristics” on page 195.
26. Added Section 22.1 “Effect of Power Reduction” on page 203.
27. Updated characteristic plots in Section 22. “Typical Characteristics” for ATtiny2313A (pages 204 - 227), and added plots for ATtiny4313 (pages 228 - 251).
28. Updated Section 4. “Register Summary” on page 7 .
29. Updated Section 6. “Ordering Information” on page 11, added the package type 20M2 and the ordering code -MMH (VQFN), and added the topside marking note.

Headquarters

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

International

Atmel Asia
Unit 1-5 & 16, 19/F
BEA Tower, Millennium City 5
418 Kwun Tong Road
Kwun Tong, Kowloon
Hong Kong
Tel: (852) 2245-6100
Fax: (852) 2722-1369

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

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

Product Contact

Web Site
www.atmel.com

Technical Support
avr@atmel.com

Sales Contact
www.atmel.com/contacts

Literature Requests
www.atmel.com/literature