AVR Programming with the ATtiny45

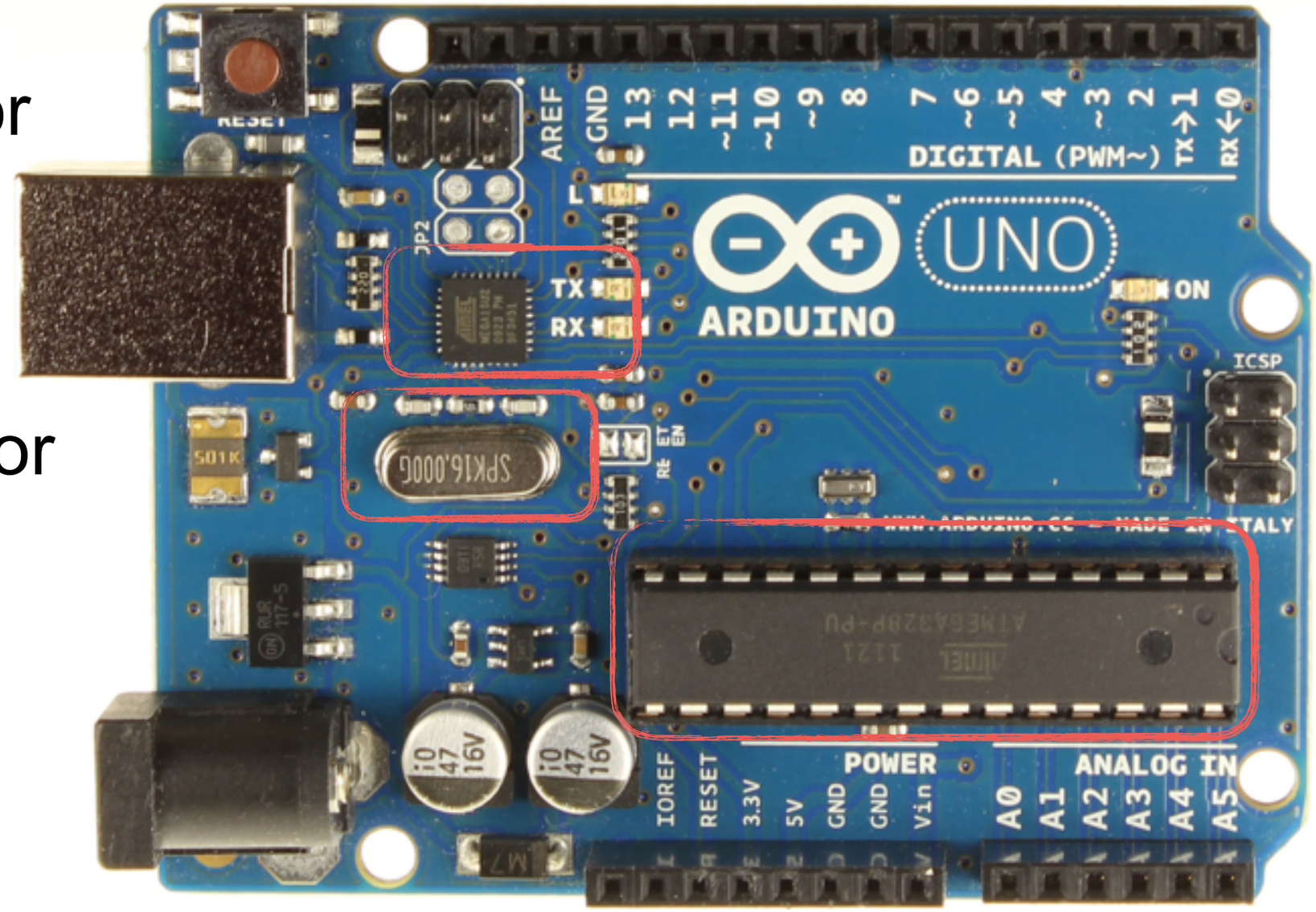
Becky Stewart and Ingo Randolf

What is a microcontroller?

- Processor, memory, some other useful stuff
- Arduino is a microcontroller platform in that it contains a micrcontroller and a bunch of other things to make interacting with that micrcontroller easier

What is on the Arduino Uno board?

- microcontroller (ATmega328P) with bootloader
- voltage regulator
- USB interface
- external oscillator

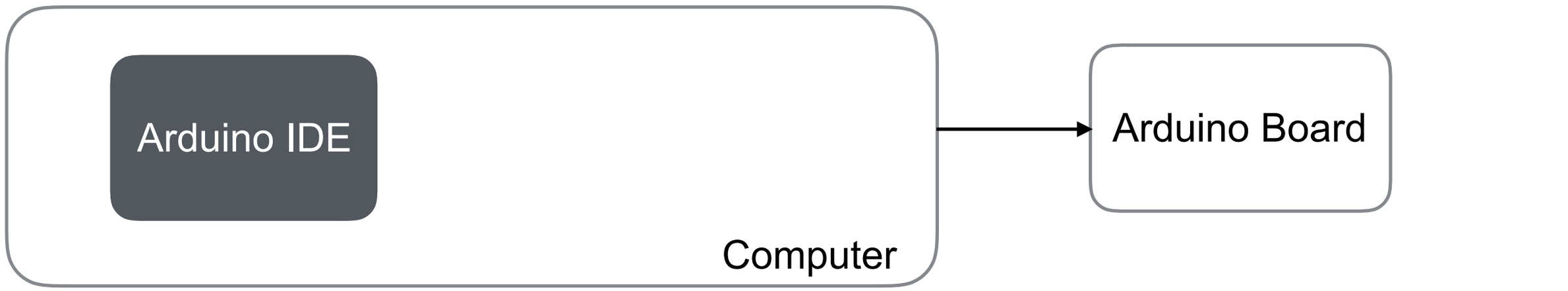


Why not just use an Arduino?

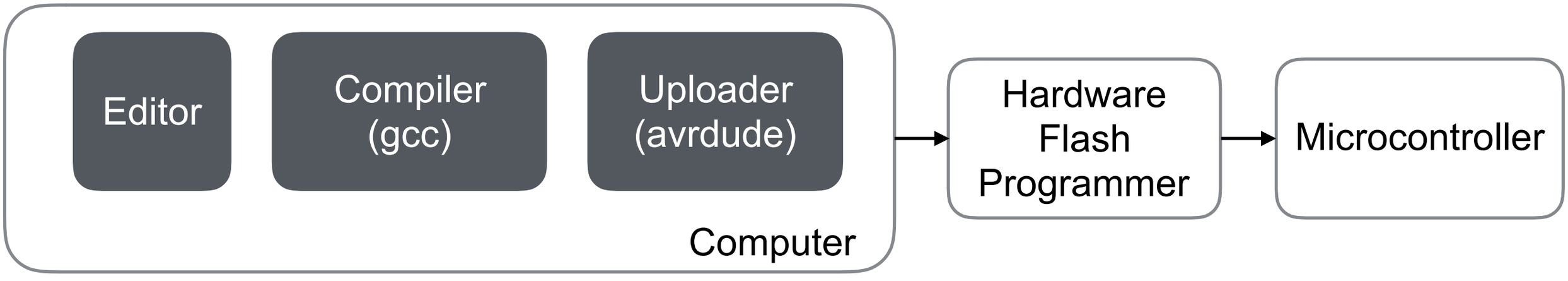
- Lots of extra stuff on there you might not need
- Cost
- It’s physically big and awkward
- Want more flexibility on pins and wiring

Toolchain

With Arduino



Without Arduino



(A Portion of the) Atmel AVR Family

ATMega

- ATMEGA168
- ATMEGA328P
- ATMEGA32U4

ATtiny

- ATTINY45-20PU
- ATTINY85V-20SU

(A Portion of the) Atmel AVR Family

ATMega

- ATMEGA168
- ATMEGA328P
- ATMEGA32U4

ATtiny

- ATTINY45-20PU
- ATTINY85V-20SU

Name of product family

(A Portion of the) Atmel AVR Family

ATMega

- ATMEGA168
- ATMEGA328P
- ATMEGA32U4

Size of Memory in KB

ATtiny

- ATTINY45-20PU
- ATTINY85V-20SU

(A Portion of the) Atmel AVR Family

ATMega

- ATMEGA168
- ATMEGA328P
- ATMEGA32U4

Atmel picoPower (low power consumption)

USB controller

ATtiny

- ATTINY45-20PU
- ATTINY85V-20SU

Max clock speed

(A Portion of the) Atmel AVR Family

ATMega

- ATMEGA168
- ATMEGA328P
- ATMEGA32U4

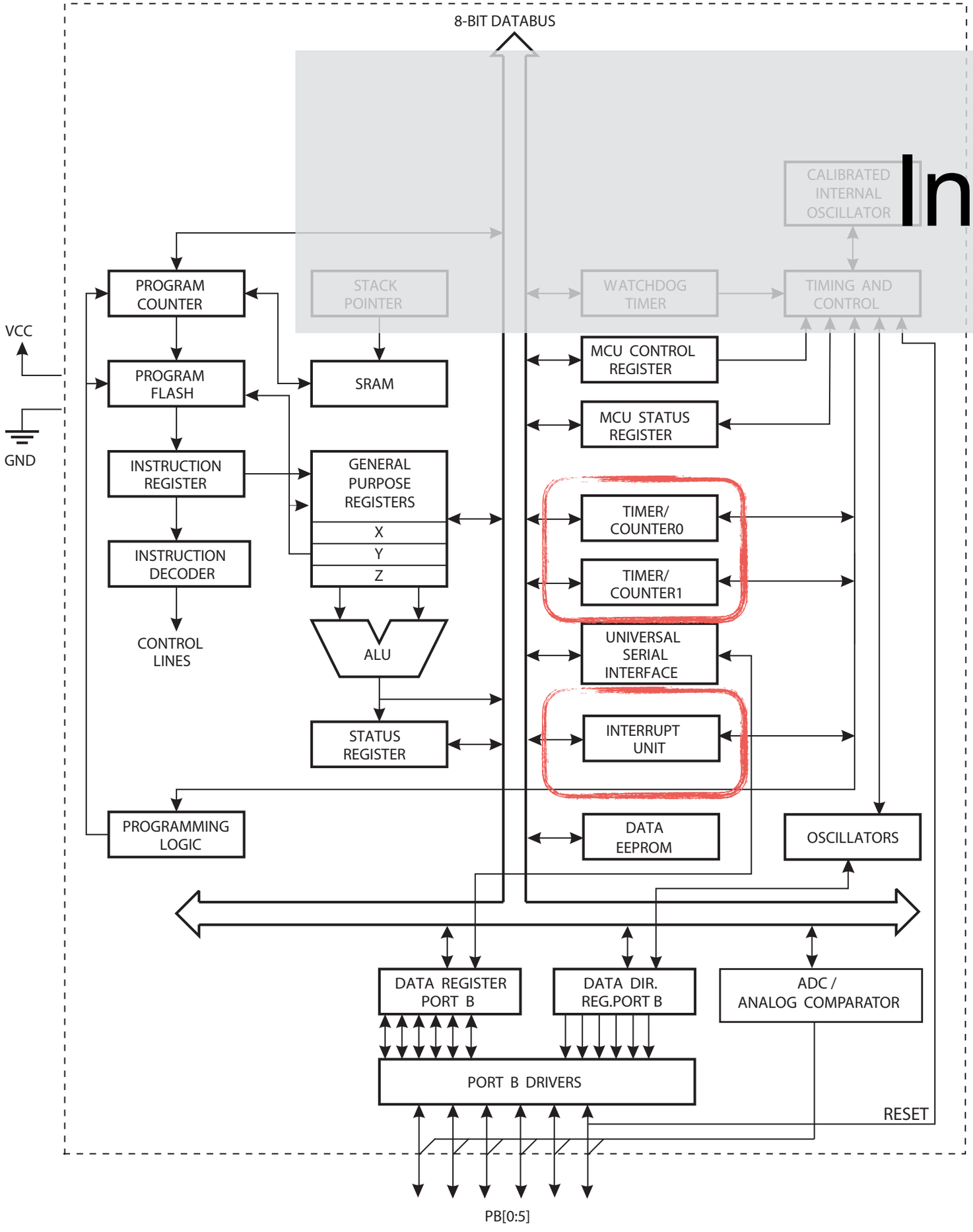
ATtiny

- ATTINY45-20PU
- ATTINY85V-20SU

Package

Can run at 1.8V instead of 2.7V

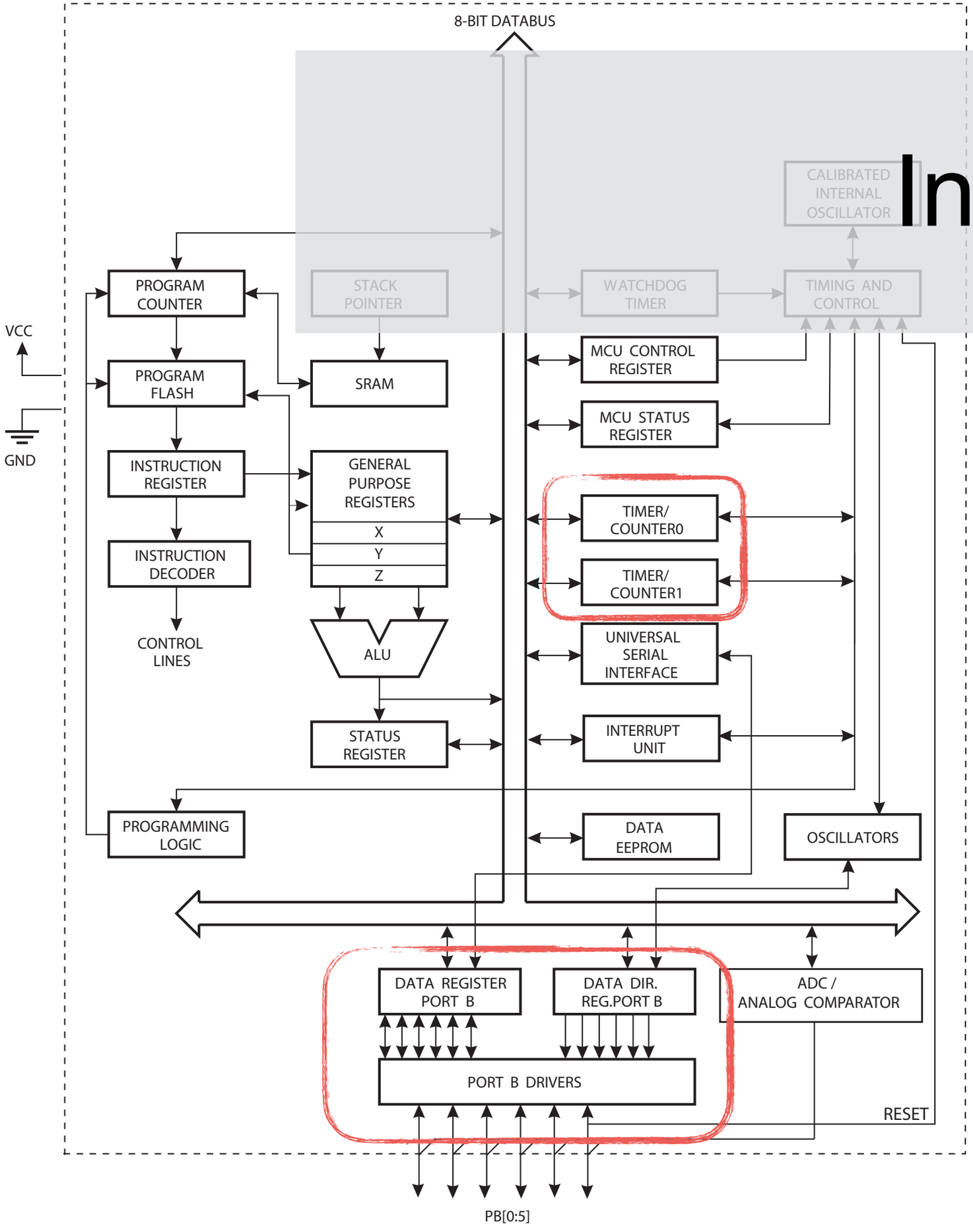
Inside the ATtiny



2 Timers/Counters

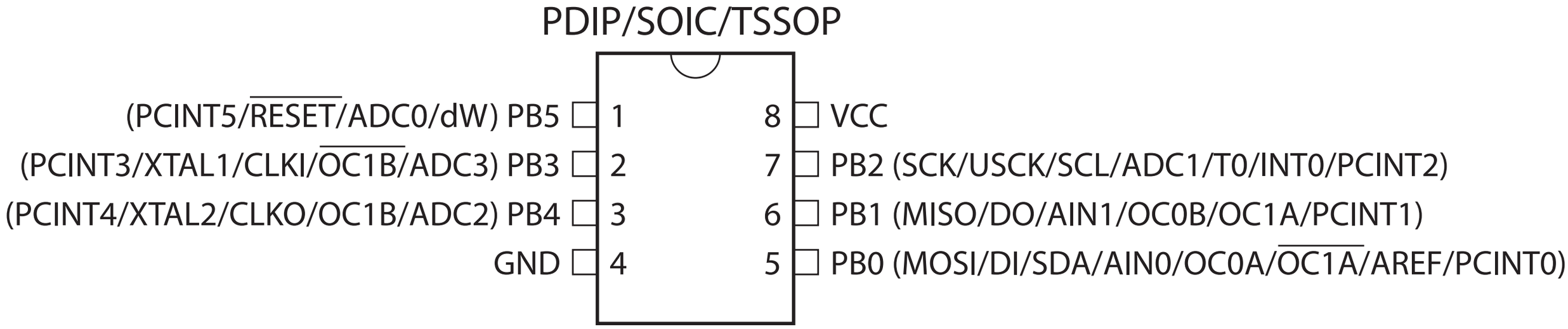
Interrupt functionality

Inside the ATtiny

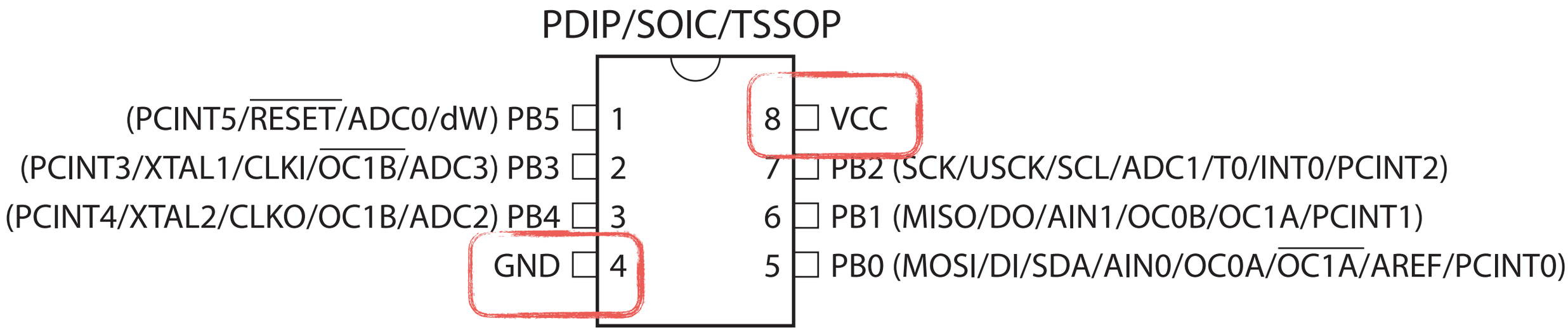


1 Hardware Register

Outside the ATtiny



Outside the ATtiny

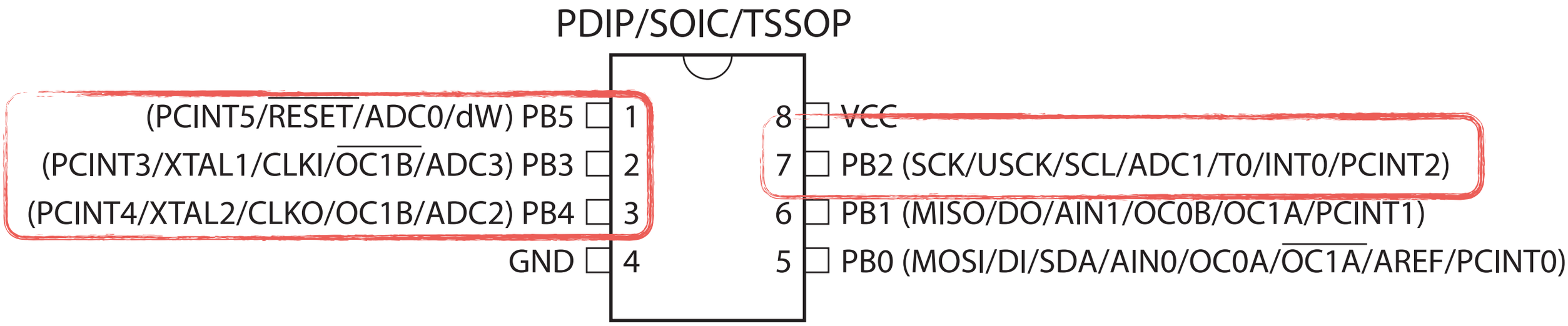


1.8 - 5.5V

or

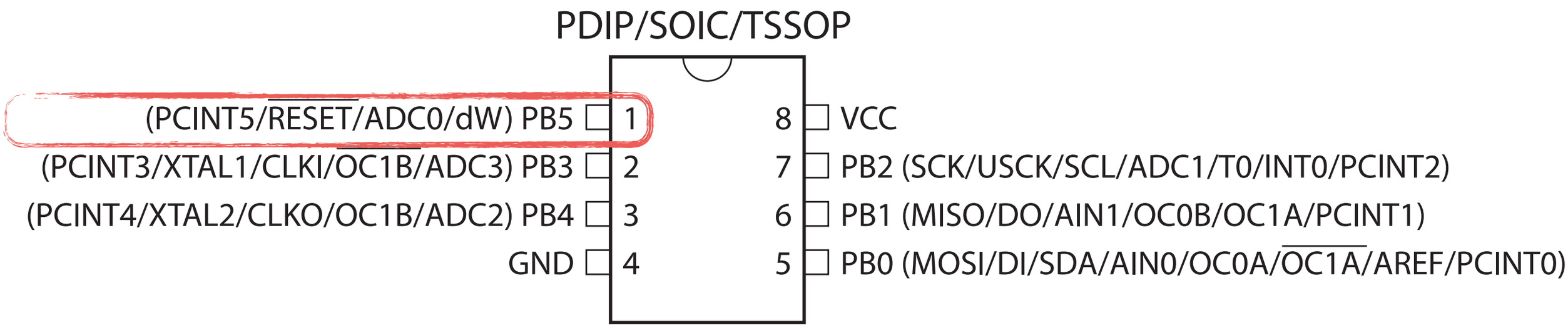
2.7 - 5.5V

Outside the ATtiny



Analog Inputs

Outside the ATtiny



If RESET is held low for long enough, will reset the chip

Using `avrdude`

- basics of command line with avrdude
- bend legs of ATtiny so fits cleanly

makefiles

- download
- go over content

Blinking

- download code
- build circuit
- flash to chip

Looking at the Source

- Preamble
- Inits
- While
- compare this to Arduino setup() and loop()

Hardware Registers

- Data direction registers
- PORT
- PIN

Task

- Change the LED pin

Reading Input

- Download code
- Flash
- Go over source, read in switch, toggle LED, compare to similar in Arduino

Reading Input

- Read in switch, toggle LED
- Read in potentiometer to set brightness of LED

Blinking (Reprise)

- blinking without delay with variables
- blinking with timer interrupt

PWM Synth with Timers

- CTC timer

External Interrupts

- pin change interrupt to trigger note?

Resources

- Make: AVR Programming by Elliot Williams
  - Example code: https://github.com/hexagon5un/AVR-Programming
- http://www.atmel.com/images/atmel-2586-avr-8-bit-microcontroller-attiny25-attiny45-attiny85_datasheet.pdf
- http://www.avrfreaks.net/
- http://www.ladyada.net/learn/avr/index.html