UNIVERSITY OF MICROCHIP
UofM
MASTERs 2007

11099 THC

Implementing an I²C™ Thermal Controller on a PIC® Microcontroller

UNIVERSITY OF MICROCHIP UOFM MASTERS 2007

Learning Objectives

- At the end of this class you should be able to:
  - Explain 3-wire fan control
  - Explain how thermal controllers work
  - Implement an I²C™ Thermal Controller using a PIC16F886 Microcontroller
- Prerequisites
  - Analog-to-Digital Converter
  - Compare/Capture/PWM
  - Timer 1

Agenda

Today we will discuss…

- Thermal Management Basics
- 3-Wire Fans
- Thermal Controllers
- Implementing a Thermal Controller on a PIC16F886

UNIVERSITY OF MICROCHIP
UOFM
MASTERS 2007

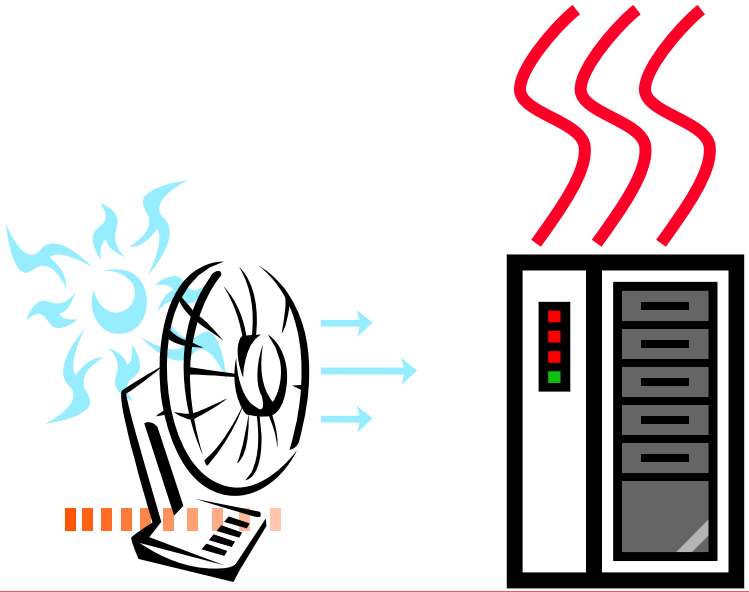
Course Background

Thermal Management

- What is thermal management?
  - Monitoring system temperature
  - Monitoring fan status
  - Optimizing the control of temperature through monitoring
  - Coordinated cooling
  - Proportional cooling

Thermal Management

- Why the need for thermal management?
  - More transistors running at higher speeds = Heat
  - Reduces system noise
  - Monitor failures
  - Lower power consumption

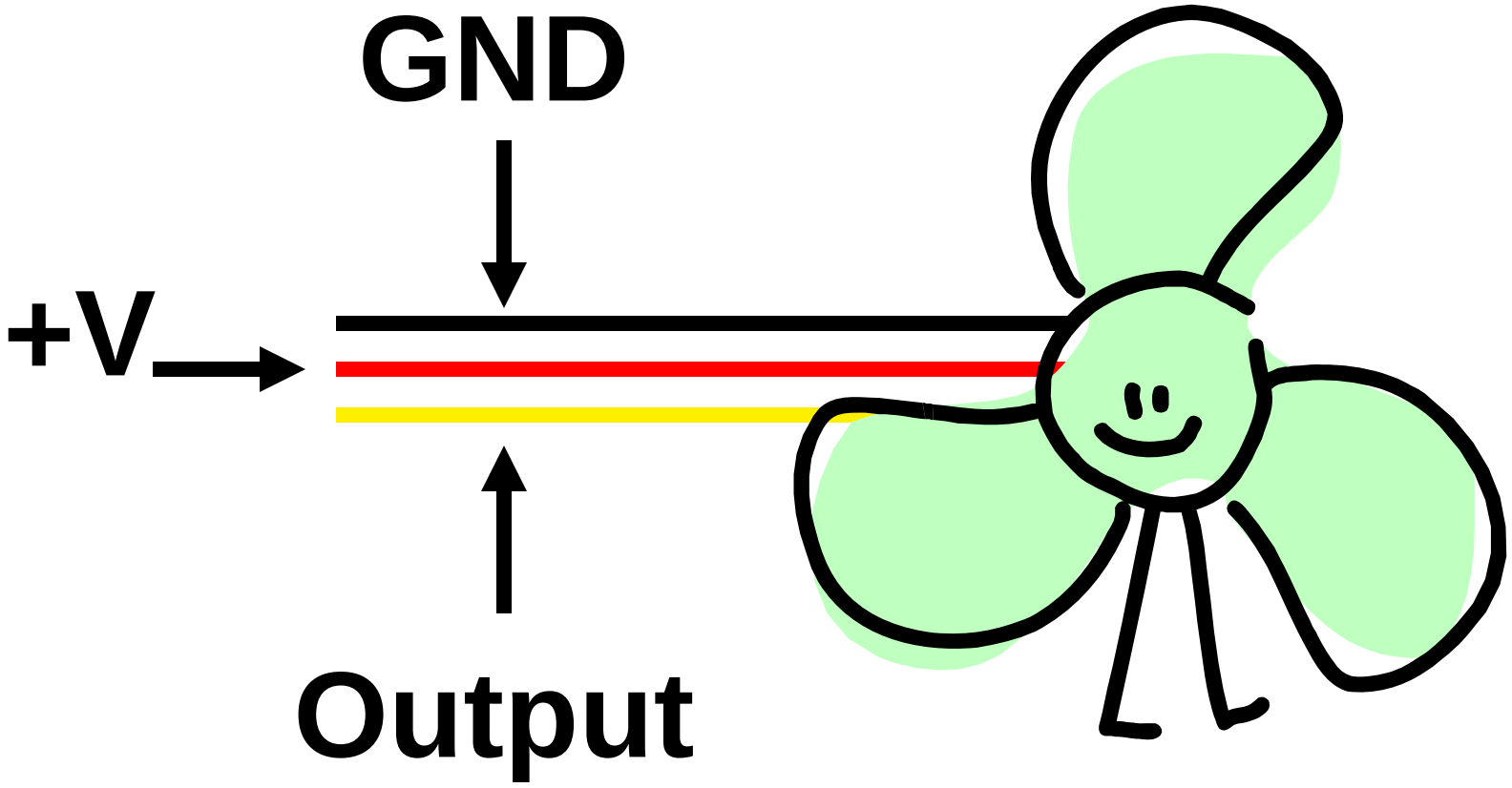


Thermal Management Controller

- Requirements:
  - Fan speed control
    - 2-, 3-, 4-wire fans
  - Temperature Sensing
  - Failure Detection
    - Fan speed detection
  - Communications
    - Serial Communications Protocols

3-Wire Fans

3-Wire Fans

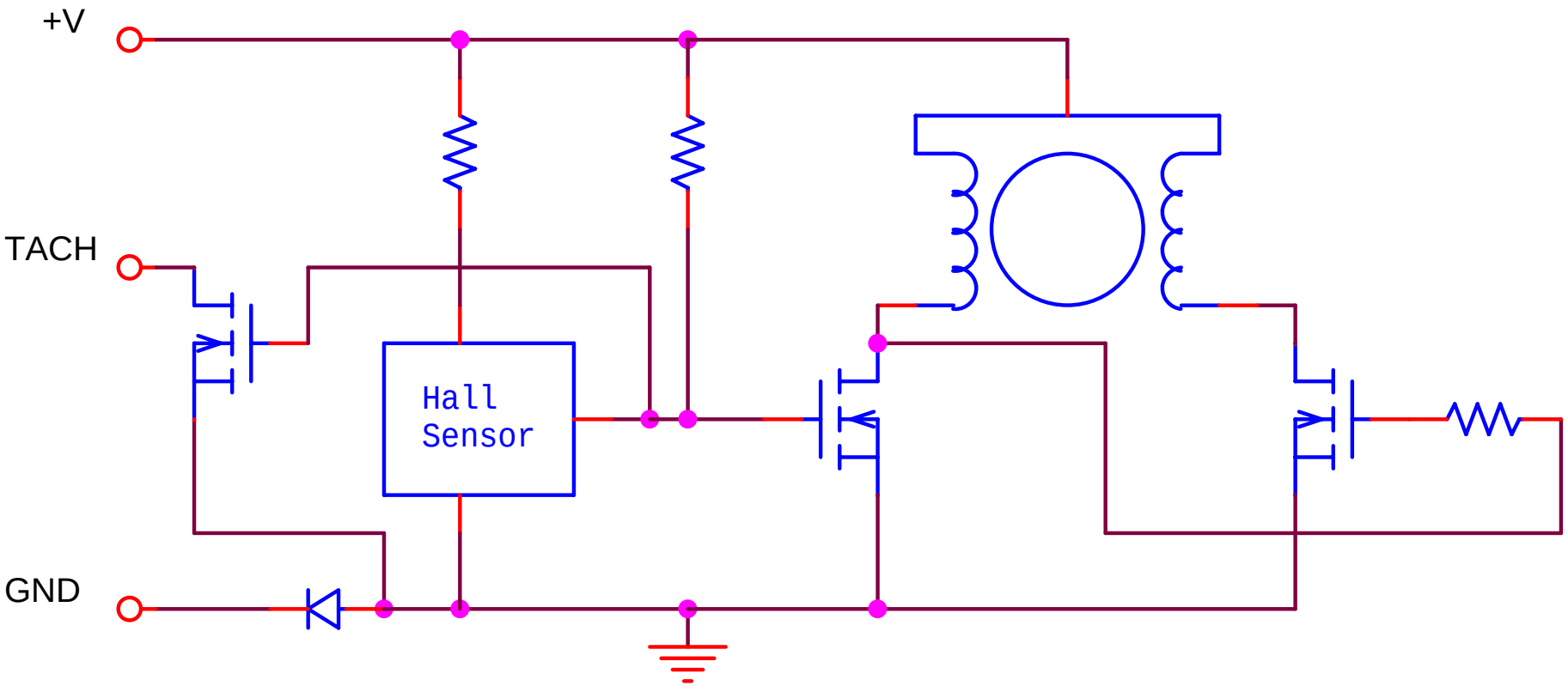


The Third Wire

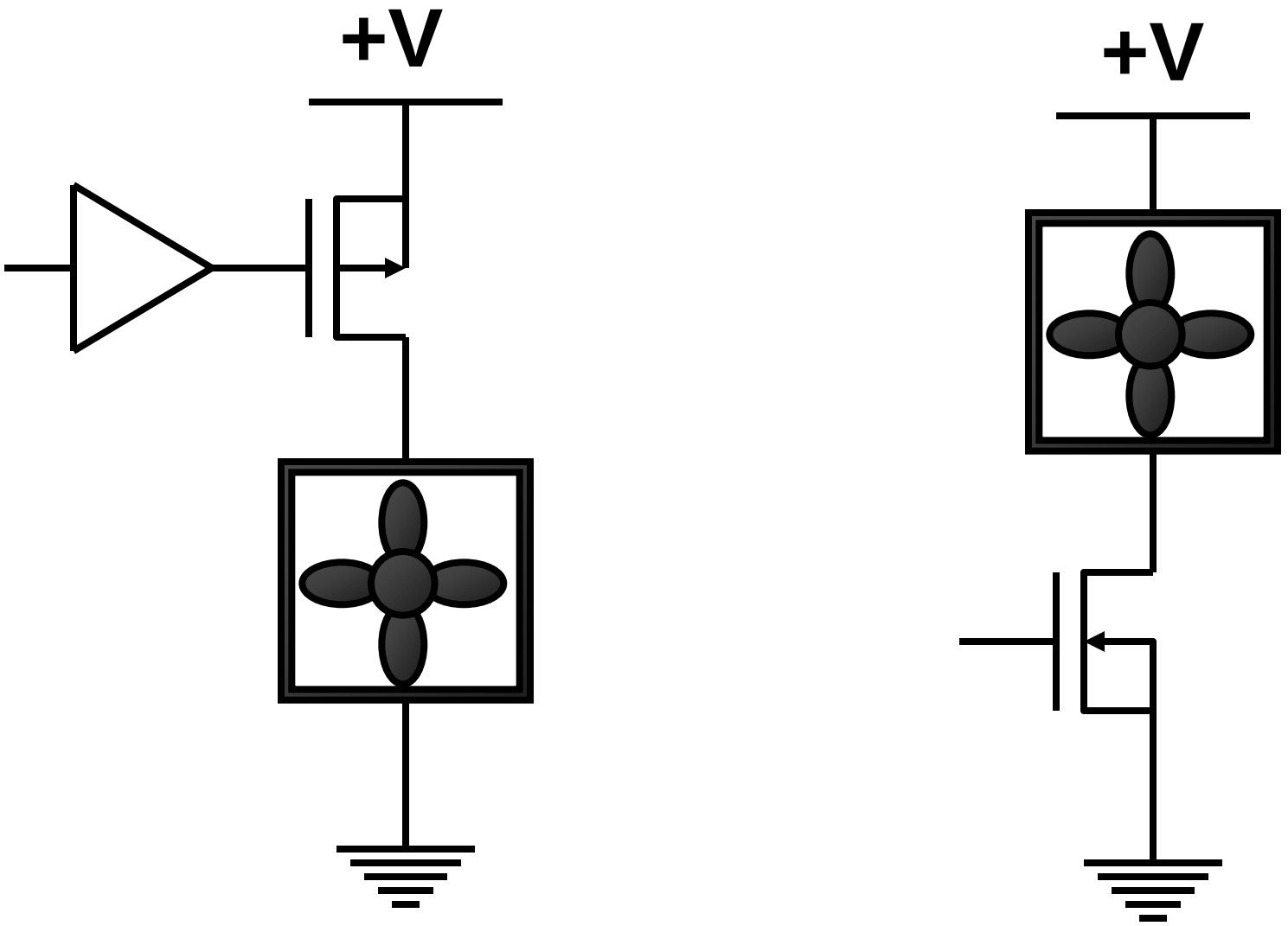
- Tachometer (most common)
  - 1, 2, or 4 pulses per revolution
  - Open collector output
- Alarm output
  - Signals when the fan has failed

3-Wire Fan Schematic

- A look inside



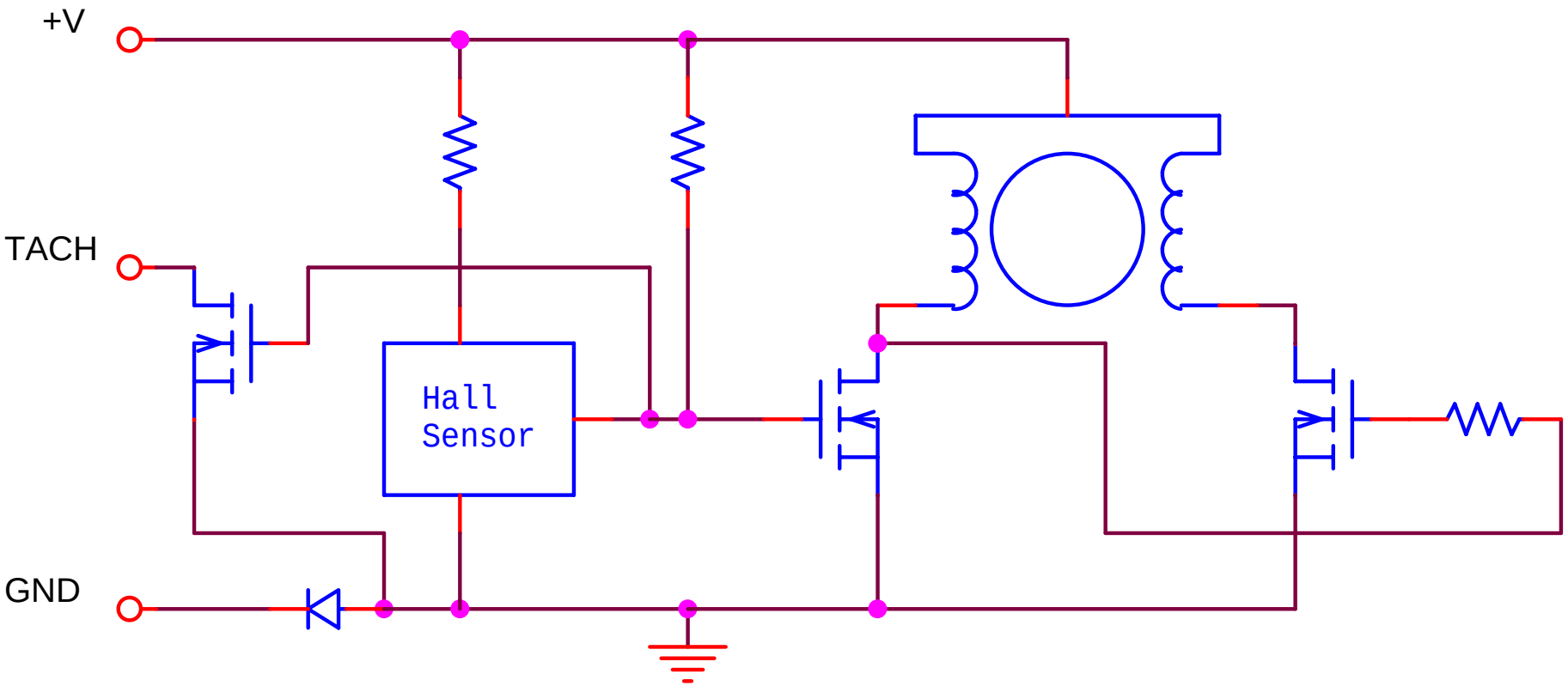
Low Frequency PWM 10-100 Hz



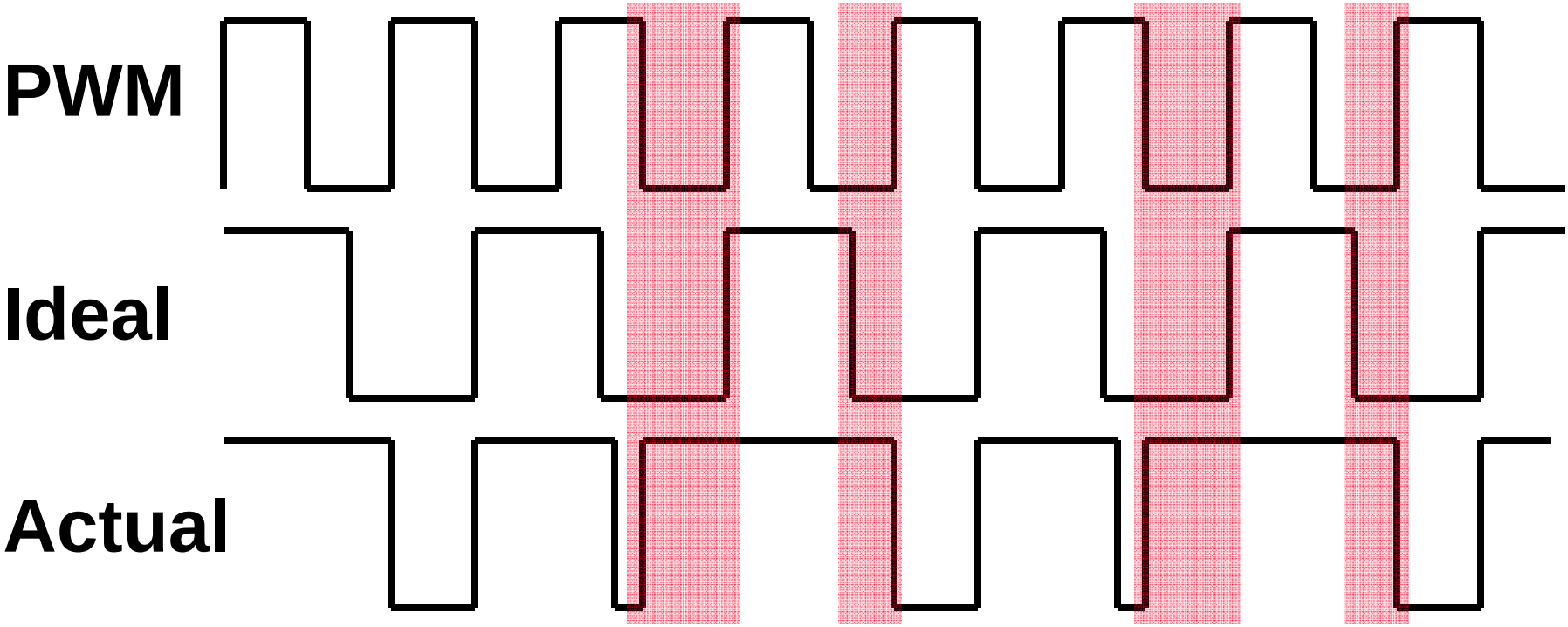
Low Frequency PWM

- Ability to extend operating range down to about 10% of max RPM
- Check with manufacturer if you can PWM the 3-Wire fan
- Tachometer output is no longer valid

3-Wire Schematic

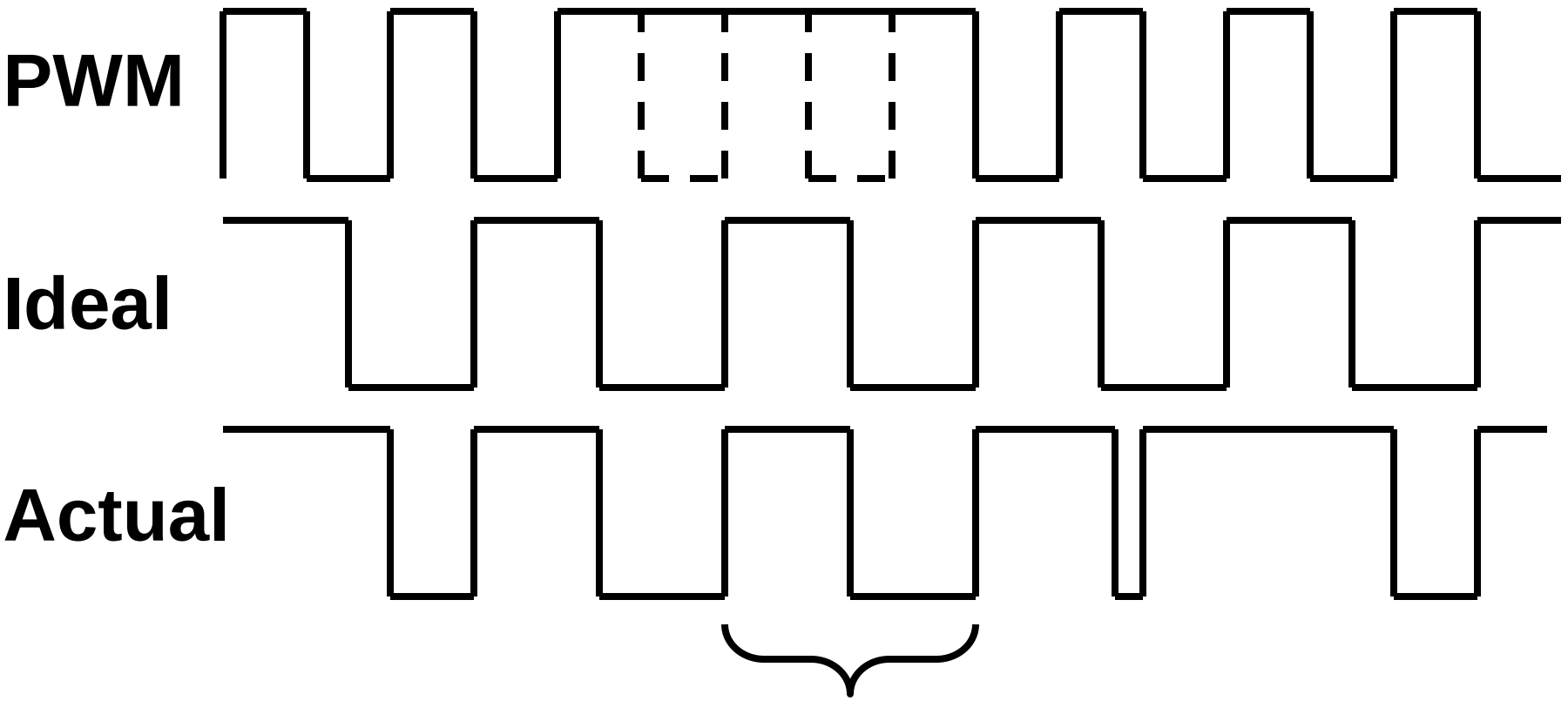


Tachometer Output



- Tachometer output is invalid

Creating a Valid Output



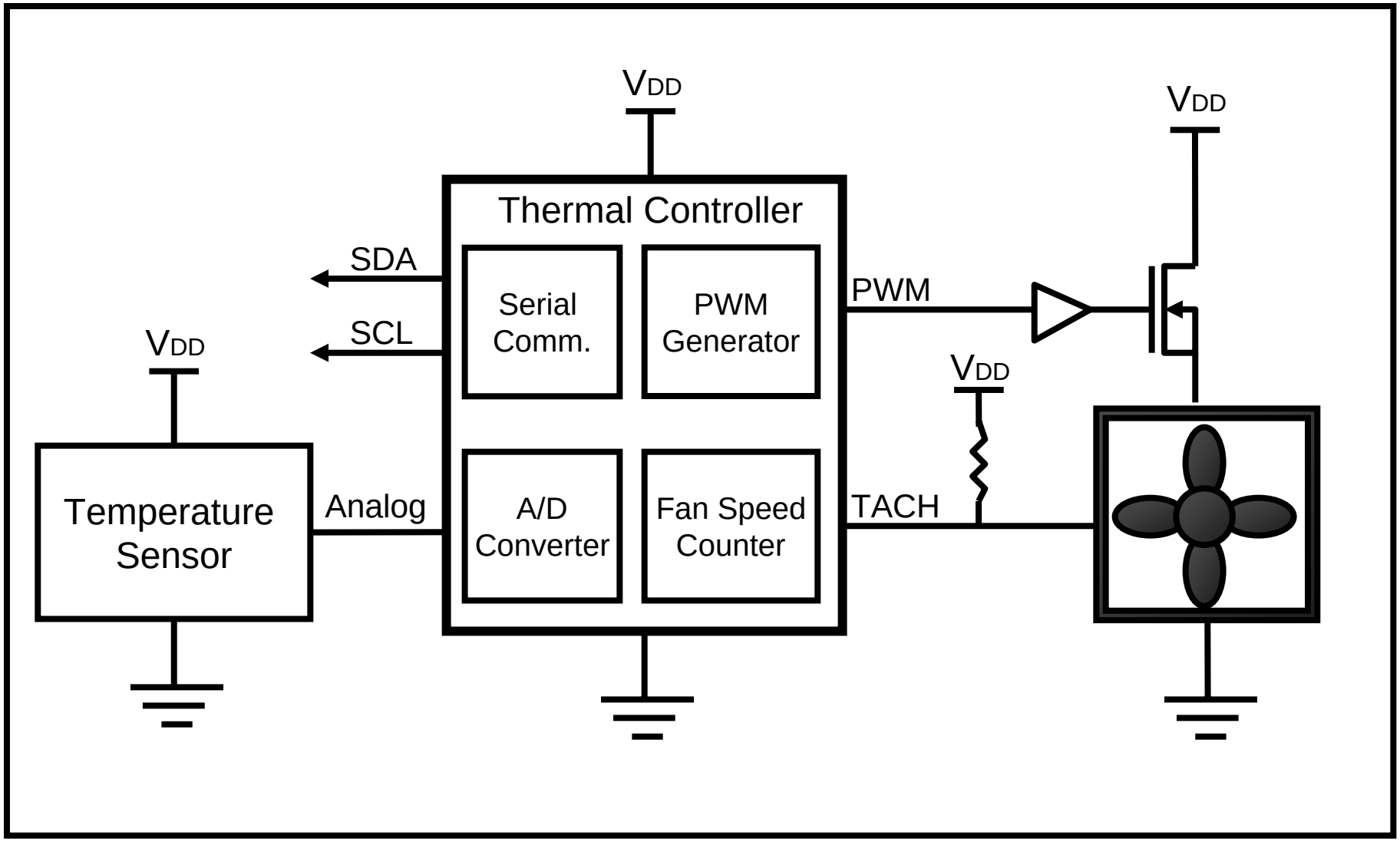
- Pulse stretching can be used to obtain a complete period

3-Wire Fans

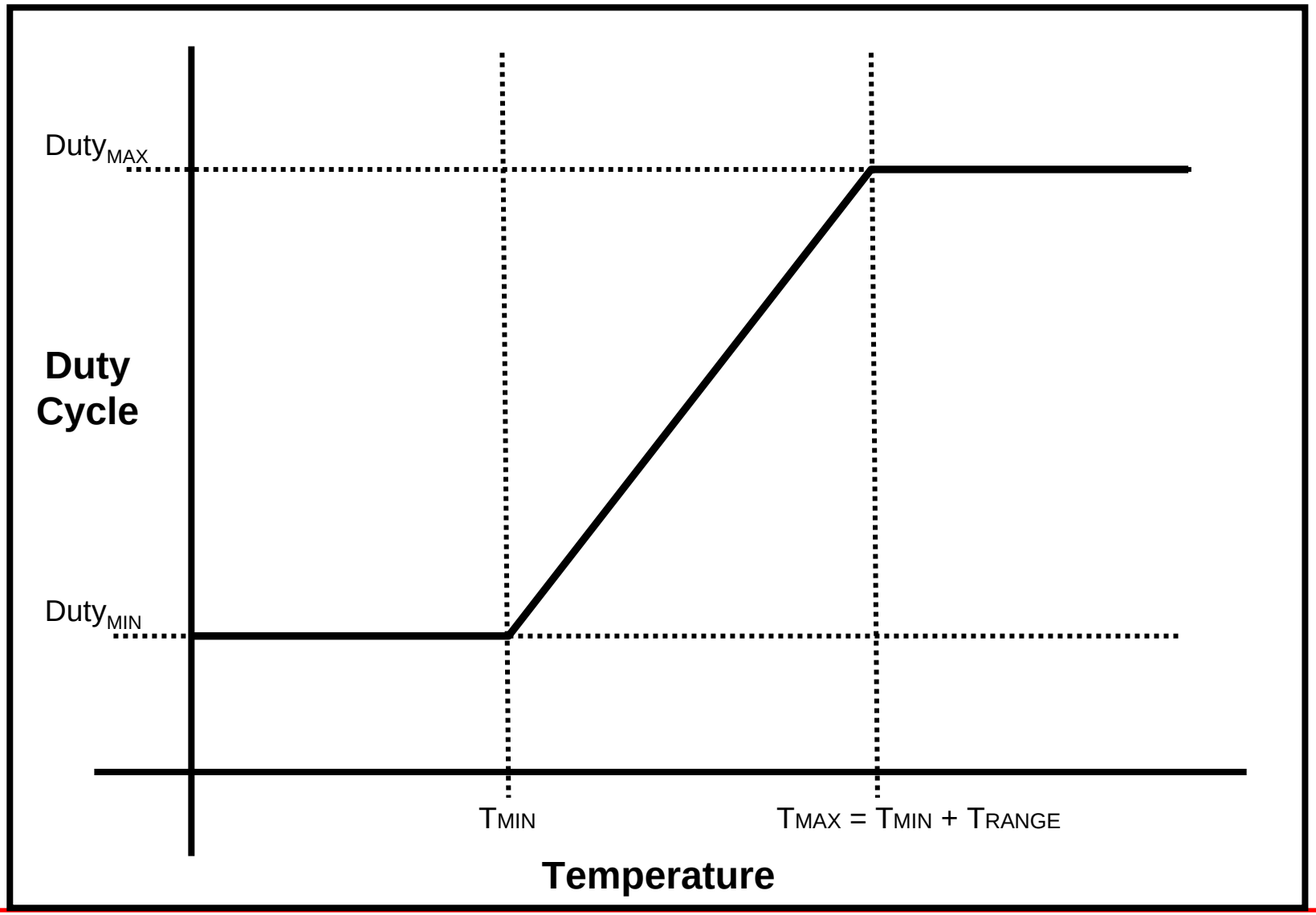
Questions?

Thermal Management Controllers

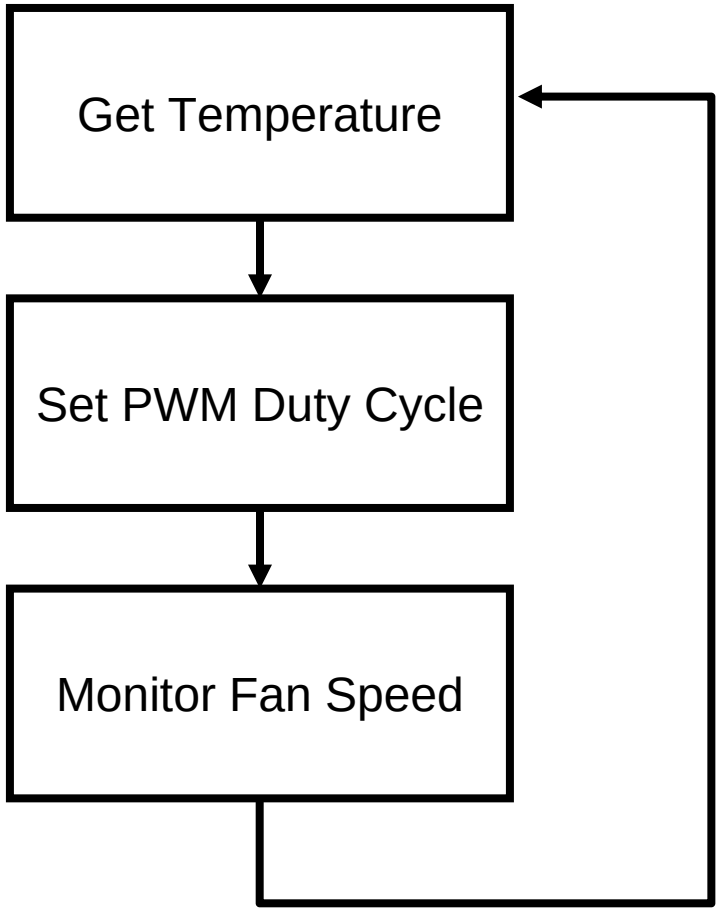
Example Thermal Controller Functions



Fan Temperature Control Algorithm



Thermal Controller Flowchart



Thermal Controller Configuration

- **Serial Communications**
- Control Registers:
  - Temperatures
  - Duty Cycle
  - PWM Frequency
- Status Registers:
  - Fan Speed
  - Temperature

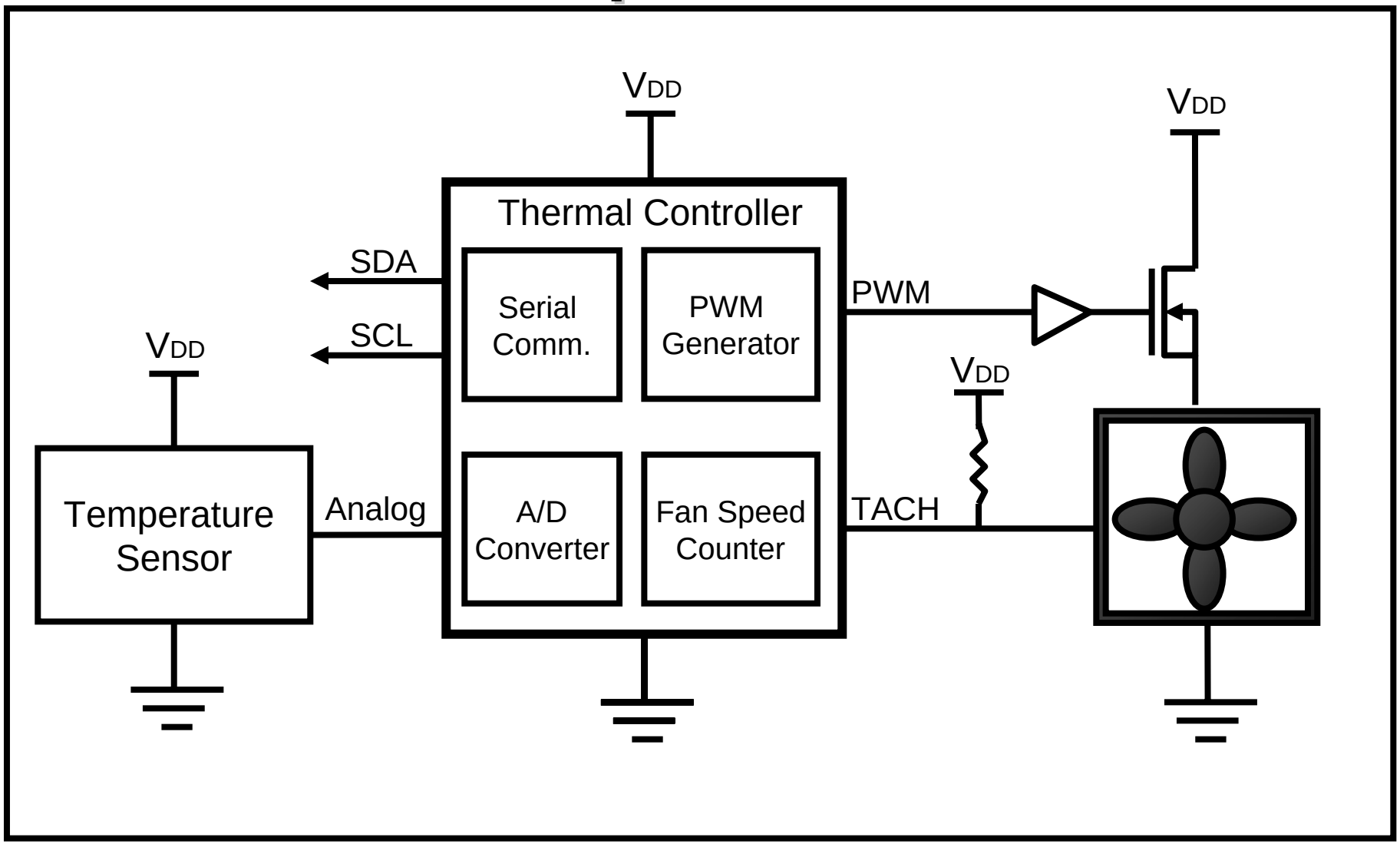
Fan Controller Registers

Configuration
Status Register
PWM Frequency
Local Temperature High
Local Temperature Low
Duty Cycle Maximum
Duty Cycle Minimum
Fan Speed Reading

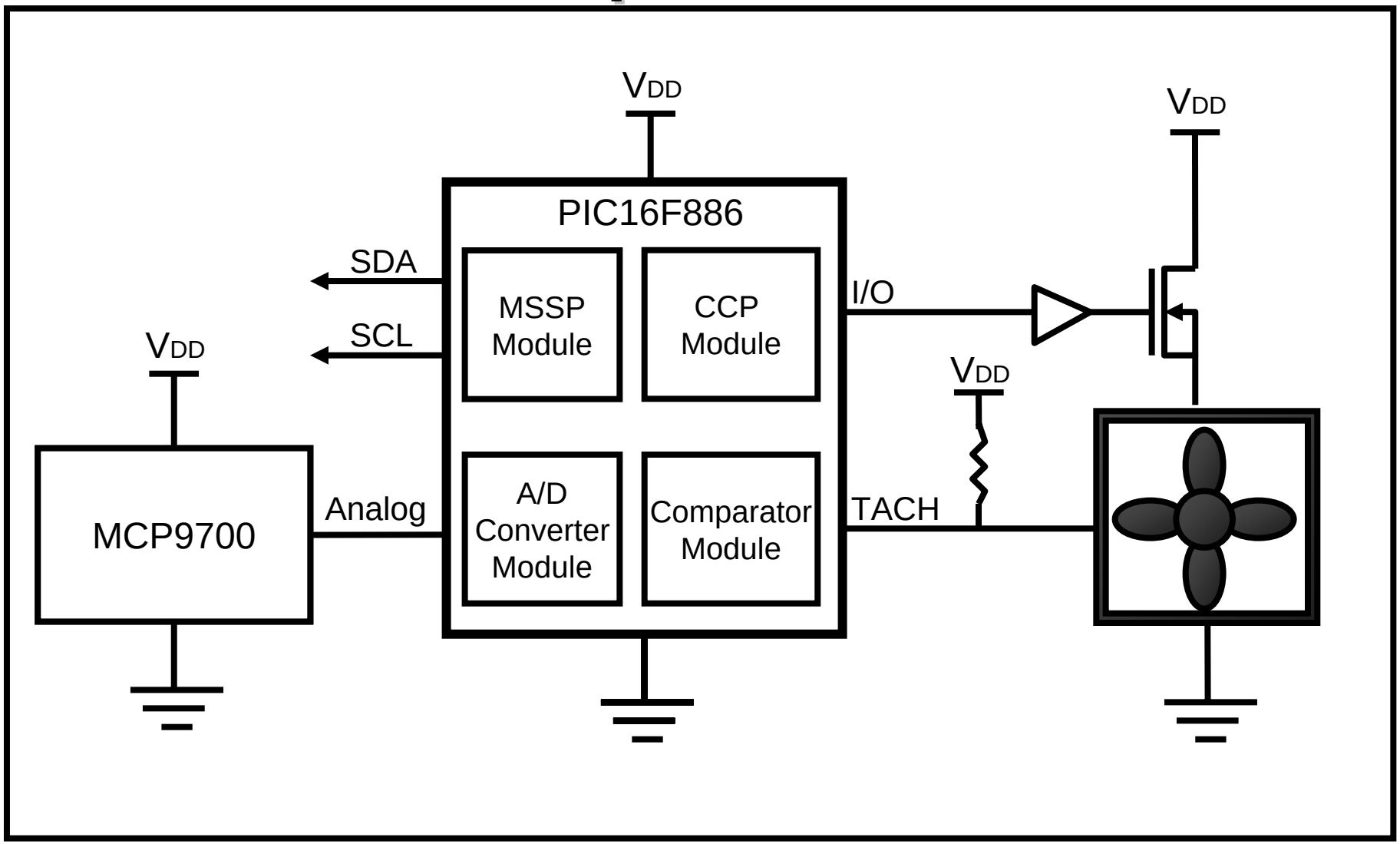
UNIVERSITY OF MICROCHIP
UOFM
MASTERS 2007

Implementing a Thermal Controller on a PIC16F886

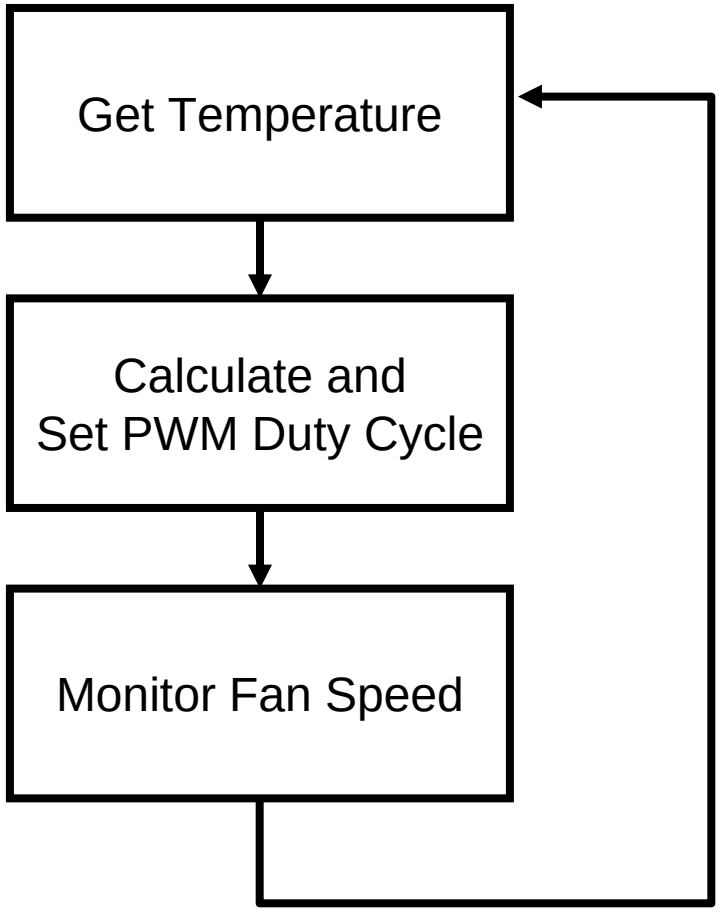
Thermal Controller Implementation



Thermal Controller Implementation

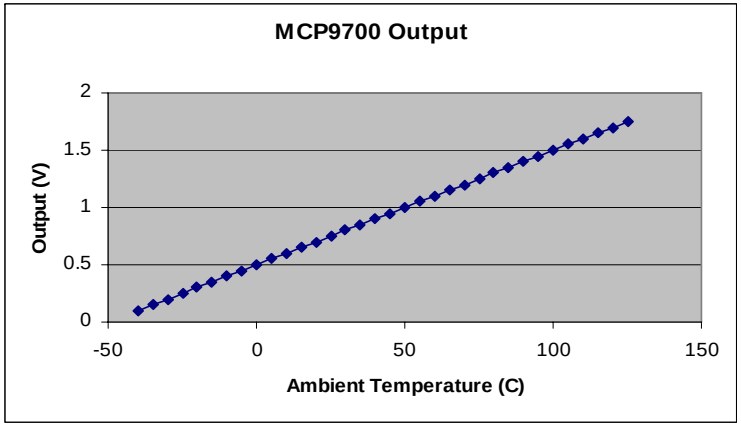


PIC® MCU Thermal Controller Flowchart



Temperature Measurement

- **MCP9700 Temperature Sensor**
- Analog Temperature Sensor



$$V_{OUT} = T_C * T_A + V_{0degC}$$

Where:

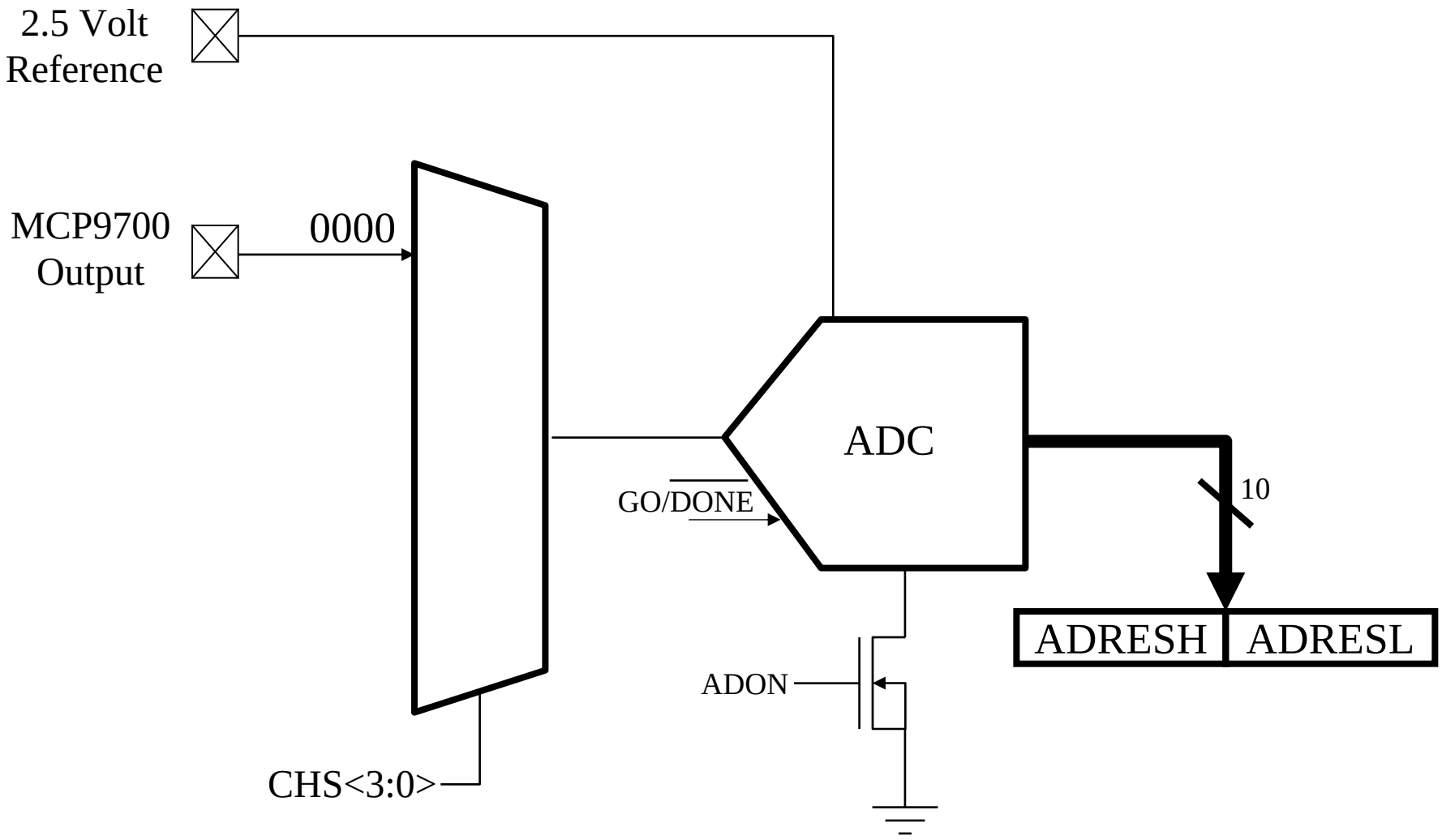
T_A = Ambient Temperature

V_{OUT} = Sensor Output Voltage

V_{0degC} = Sensor Output Voltage at 0 degC = 500 mV

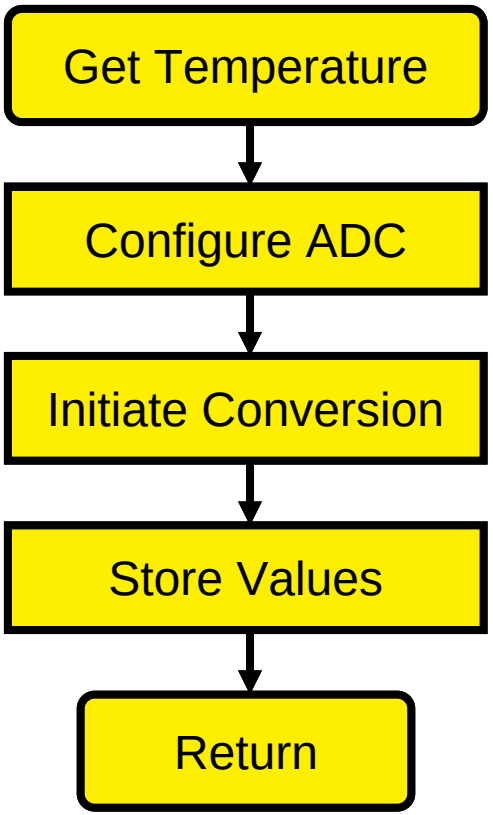
T_C = Temperature Coefficient = 10 mV/degC

Configuring Analog-to-Digital Converter

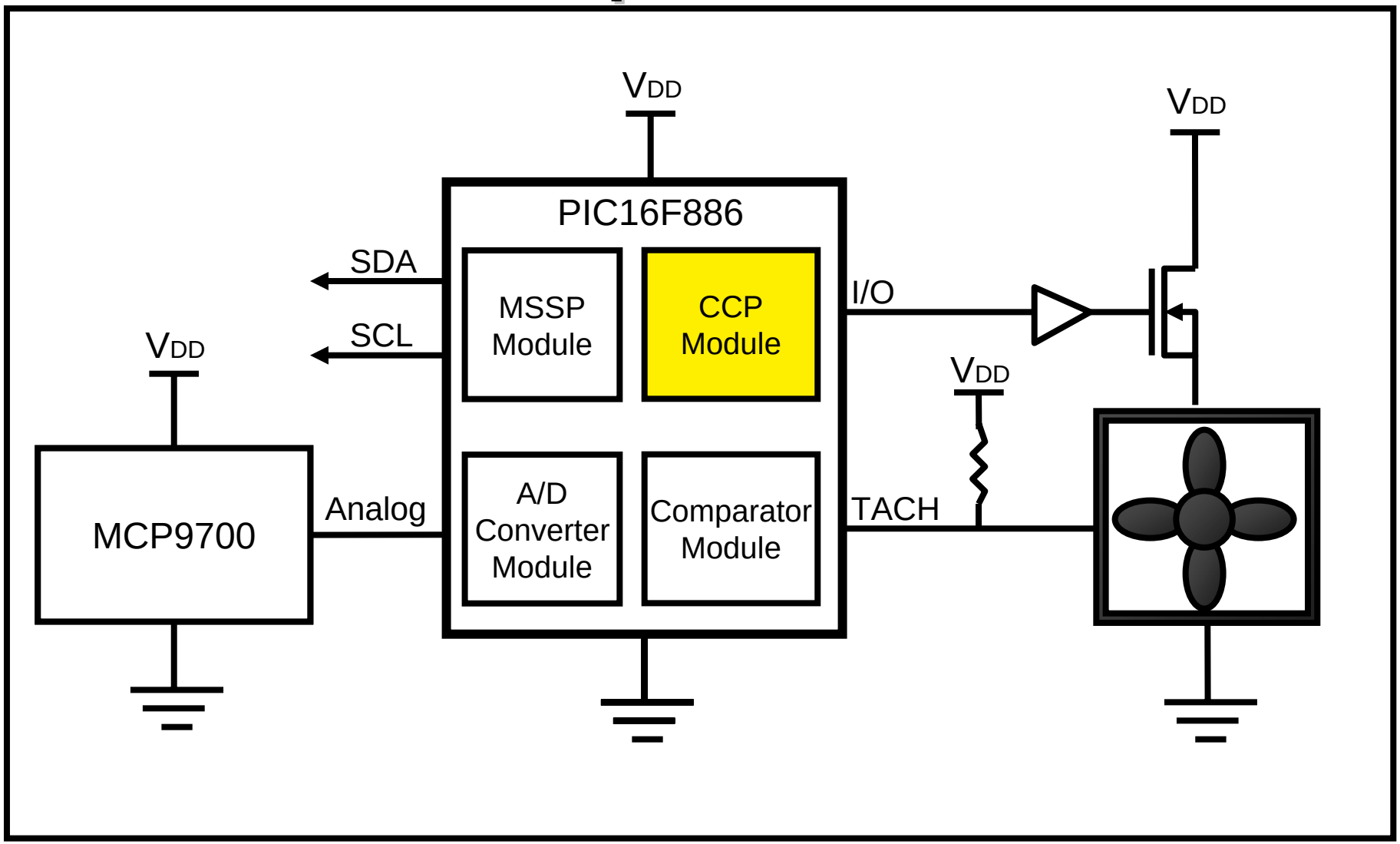


Temperature Measurement

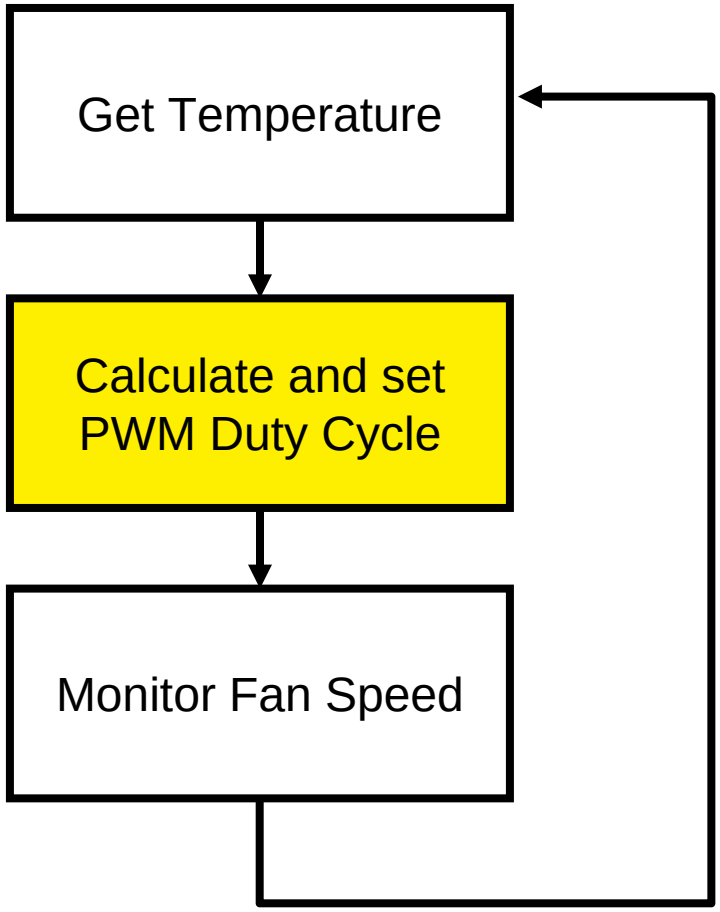
- 10-Bit Analog-to-Digital Conversion
- Store to 2 Byte Temperature Register



Thermal Controller Implementation



PIC® MCU Thermal Controller Flowchart

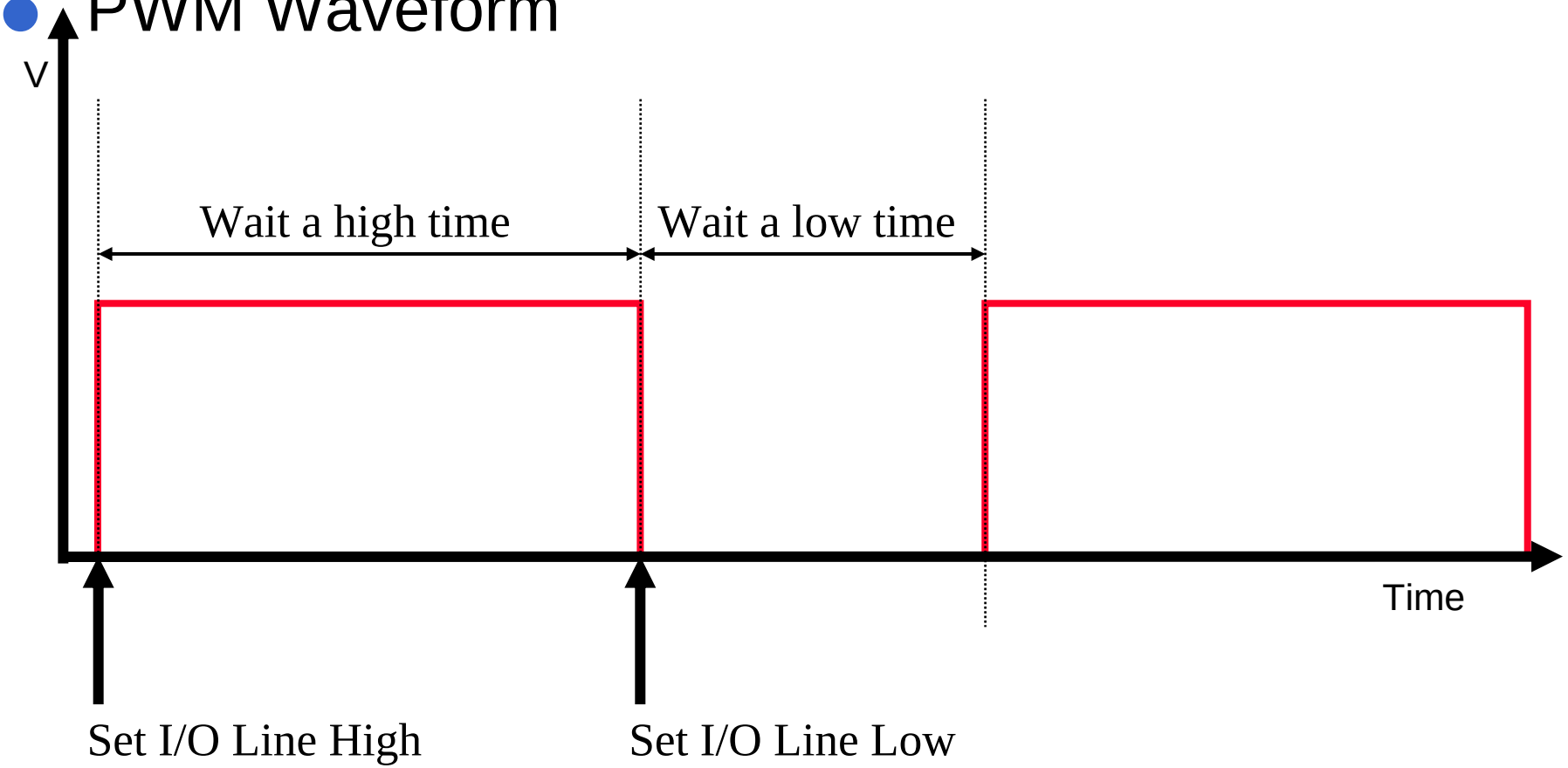


Generating a PWM Signal

- Requirement: Low-speed 10-100 Hz signal to control fan speed
- Problem: PWM module cannot generate a 10 Hz signal
- Solution: Software PWM

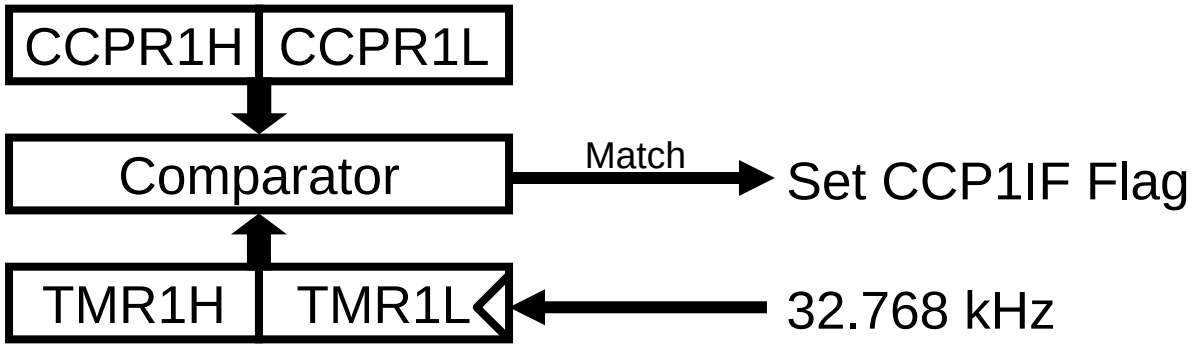
Generating a Software PWM

- PWM Waveform

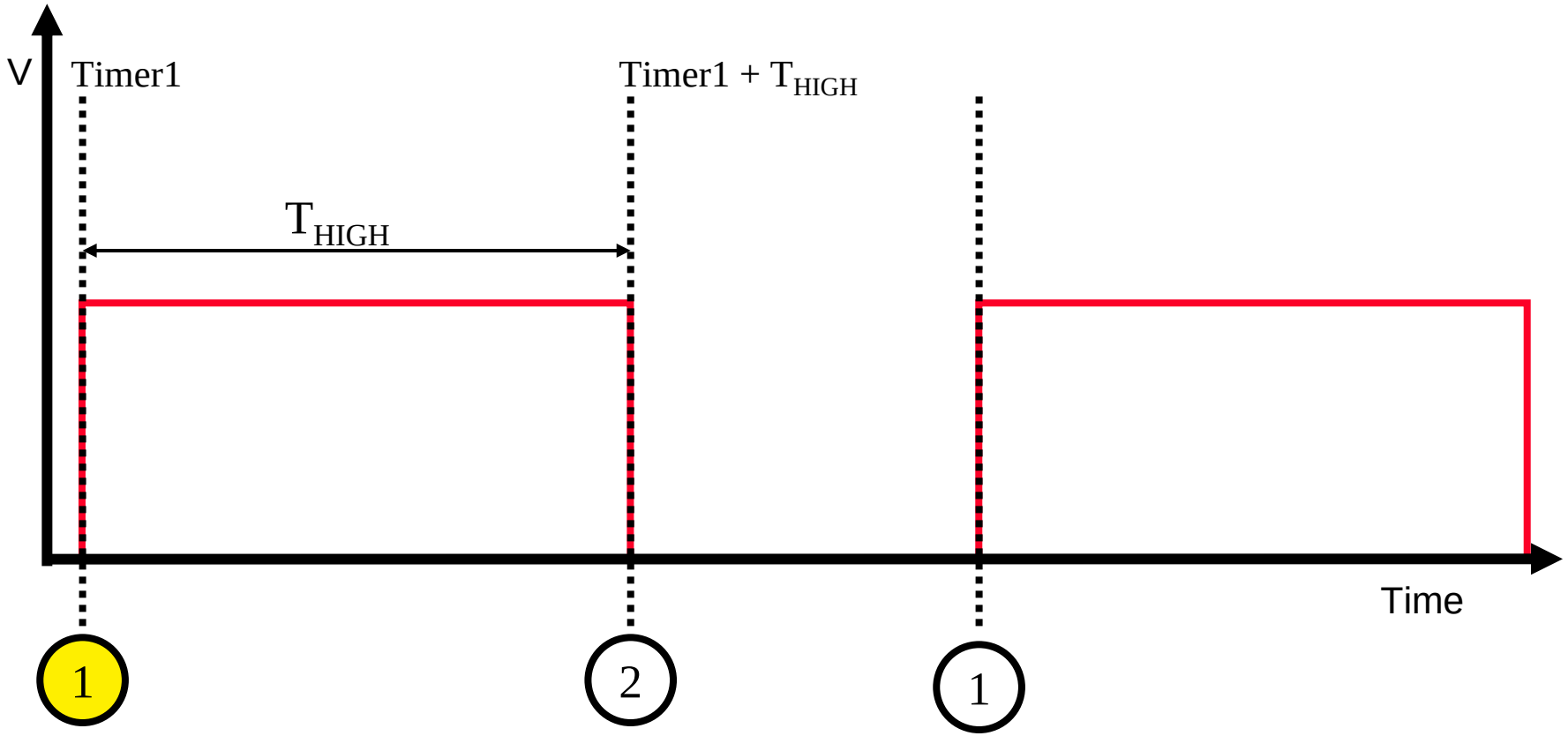


Generating a Software PWM with CCP Compare Mode

- **COMPARE** Mode to time when the I/O Line needs to be toggled
- Interrupt is generated when TMR1 equals CCPR1

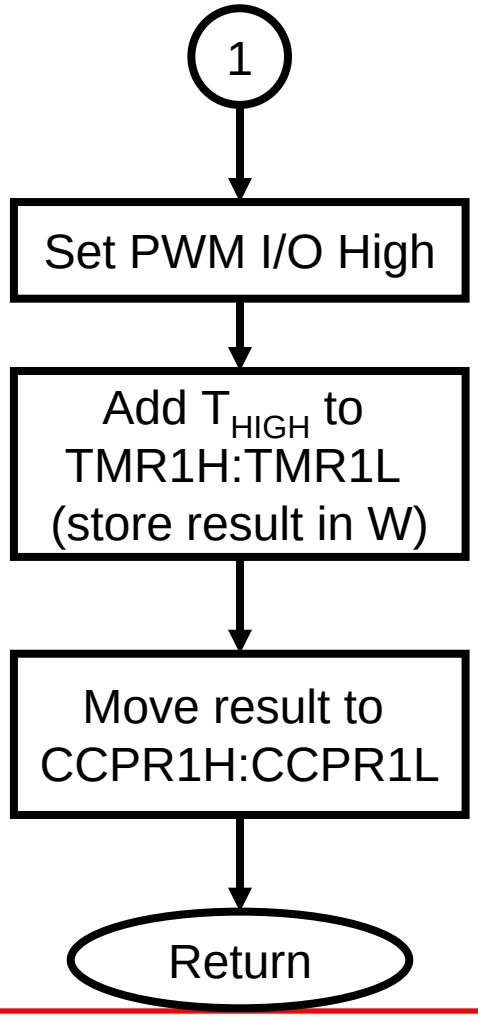


Generating a Software PWM using the CCP Compare Mode

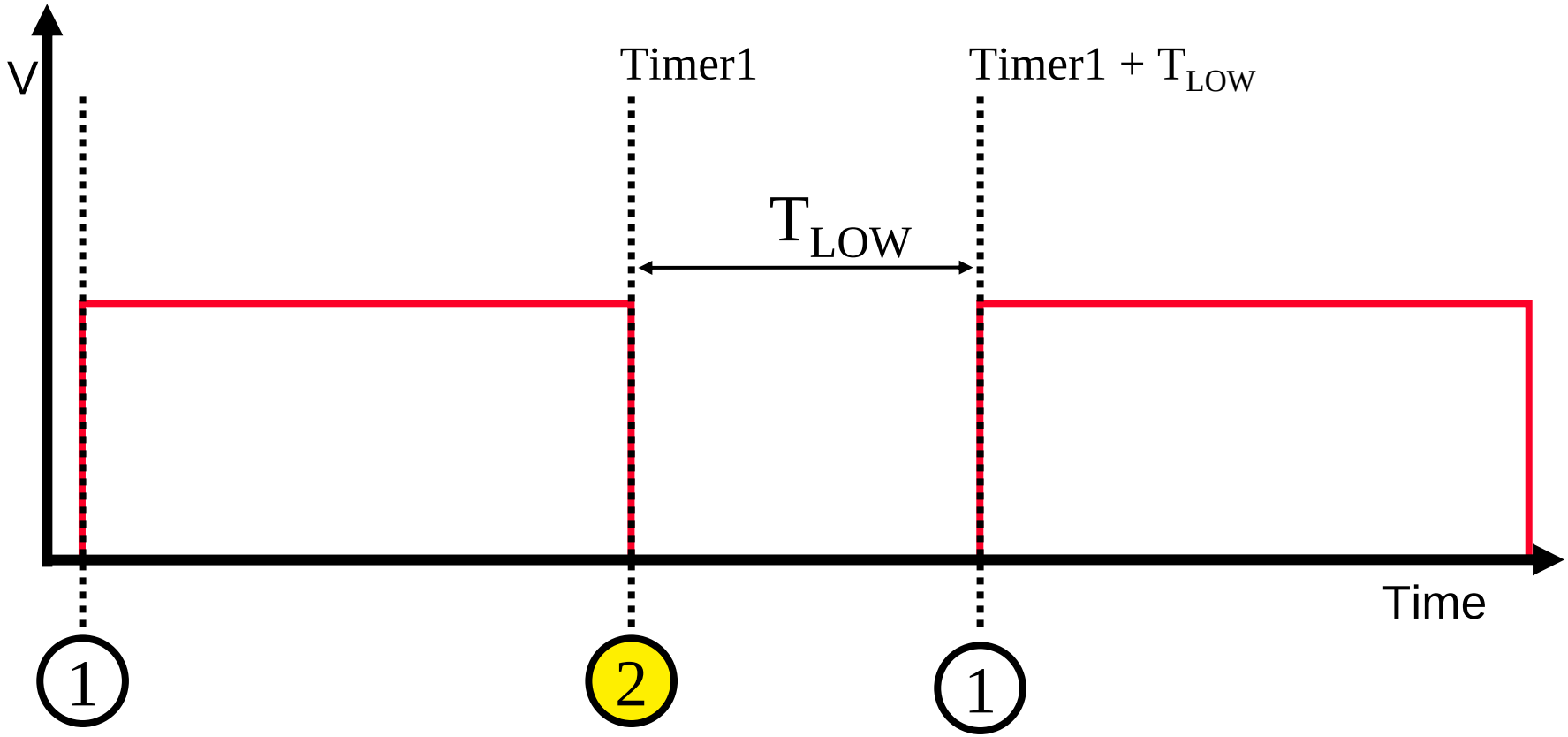


Software PWM Flowchart

- CCP (Compare) Interrupt

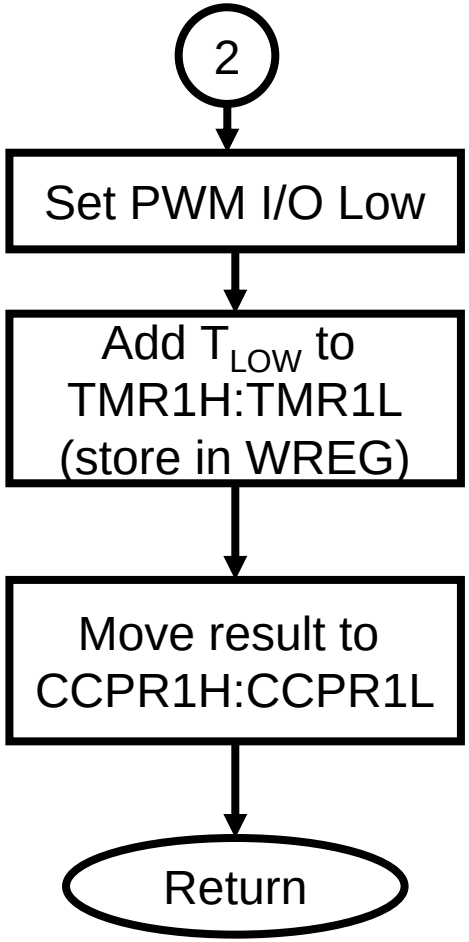


Setting PWM Duty Cycle

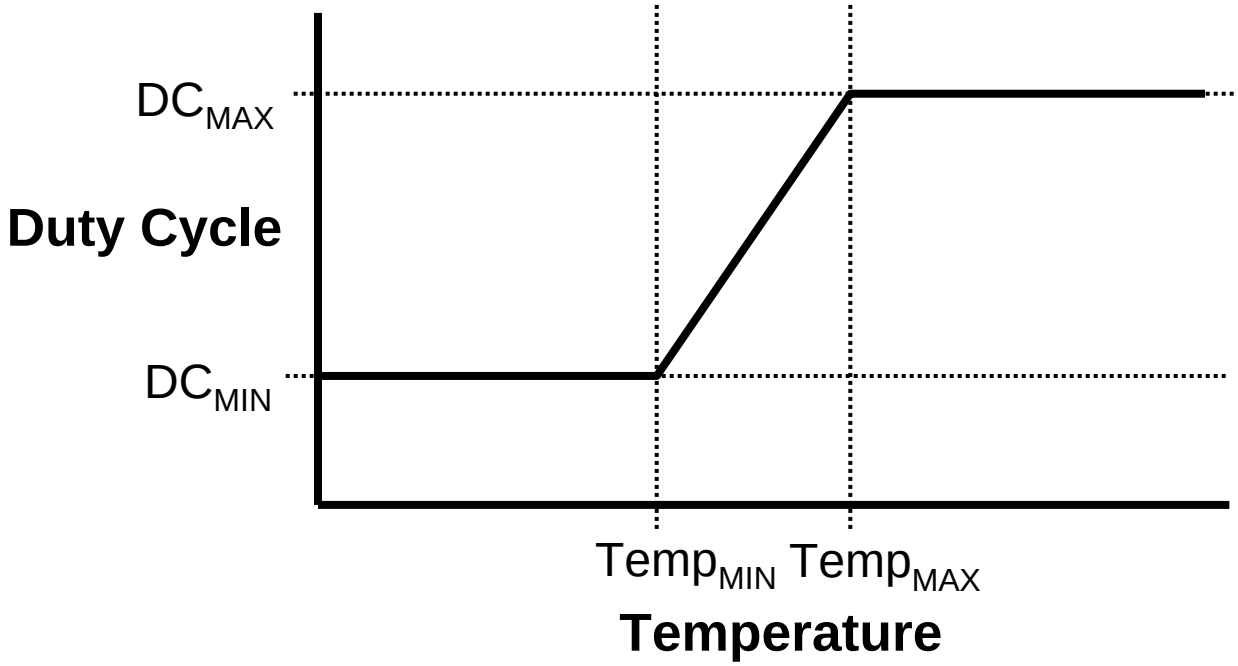


Software PWM Flowchart

- CCP (Compare) Interrupt

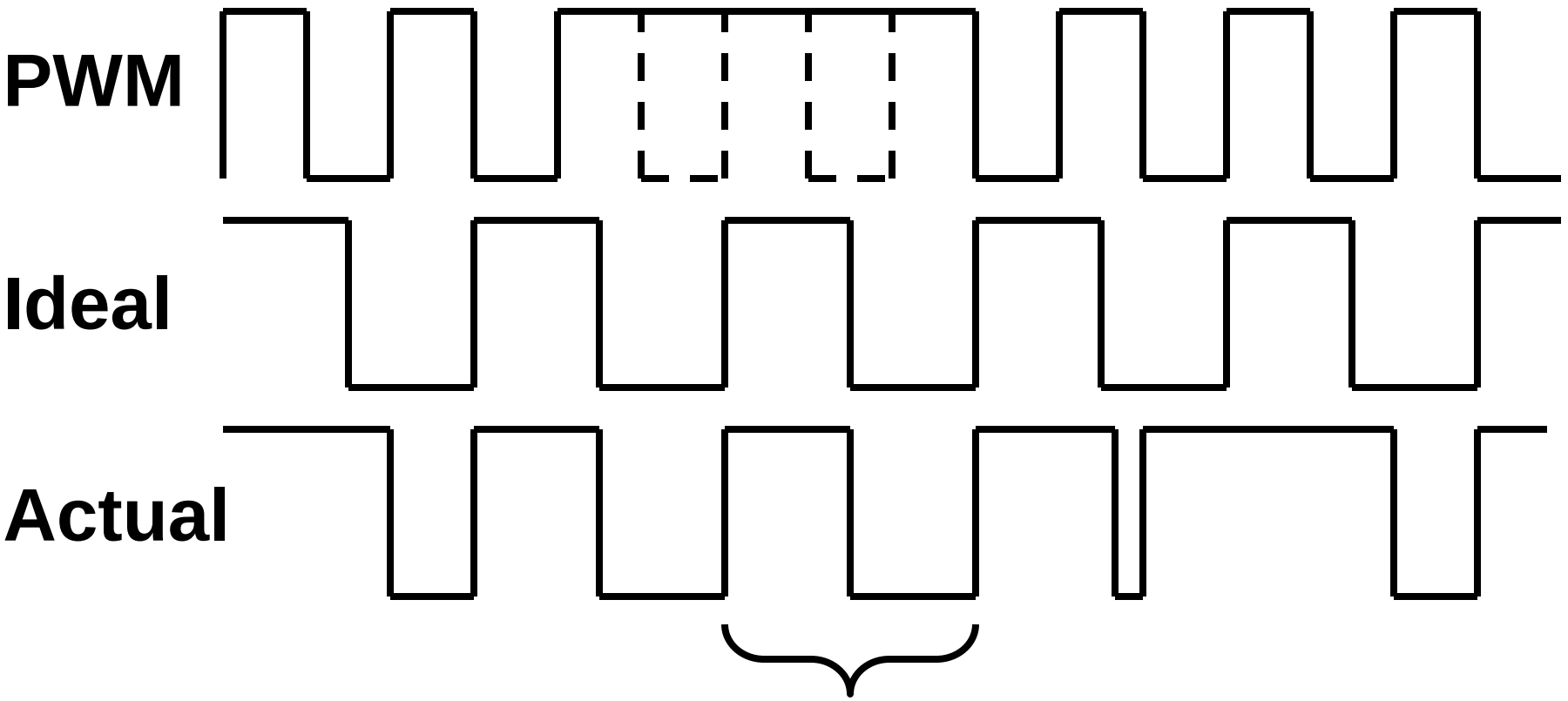


Calculating PWM Duty Cycle



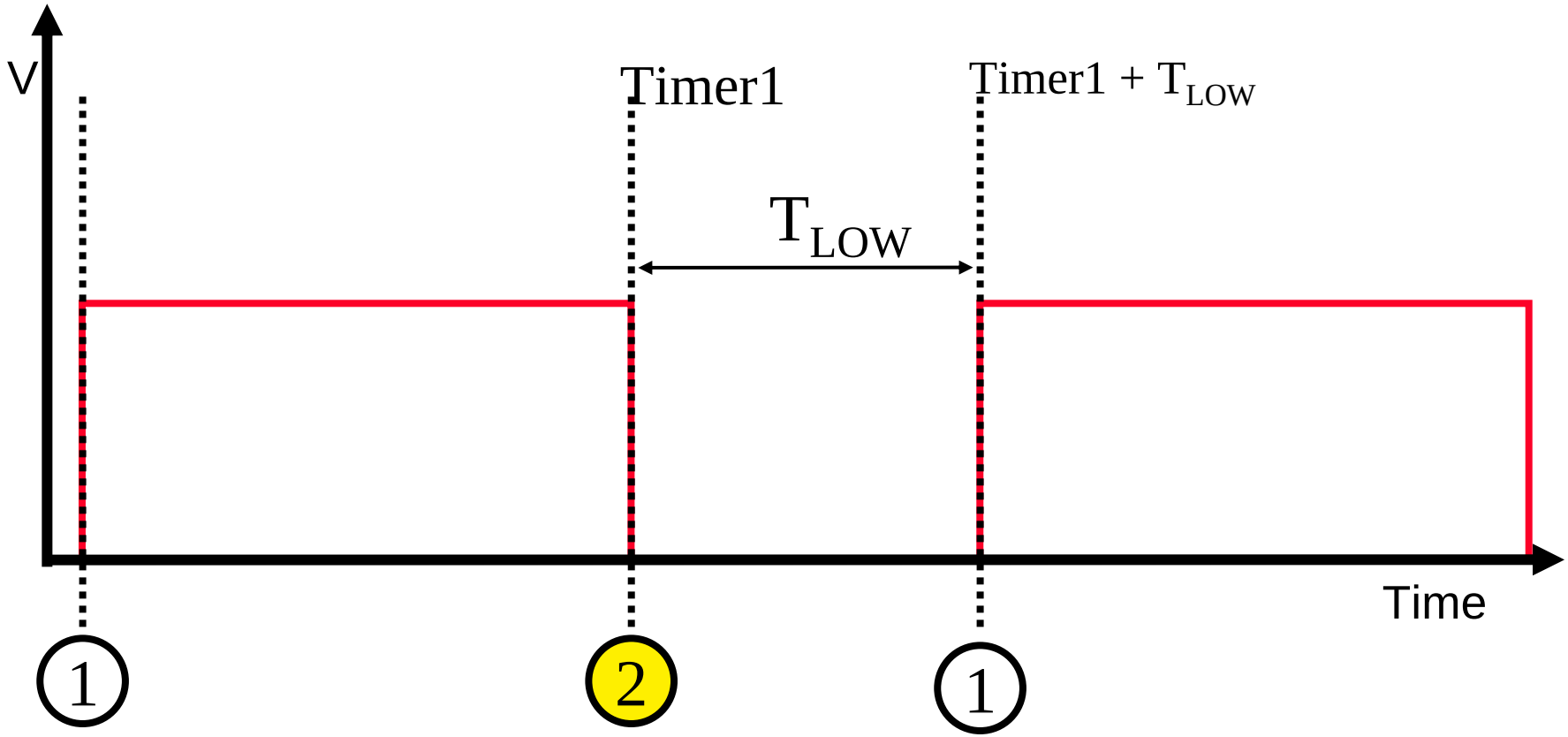
- $T_{HIGH} = \mathbf{T_A} * [(\mathbf{DC_{MAX}} - \mathbf{DC_{MIN}}) / (\mathbf{Temp_{MAX}}\text{-}\mathbf{Temp_{MIN}})]$
- $T_{LOW} = \mathbf{T_{PERIOD}} - T_{HIGH}$

Recall: Pulse Stretching



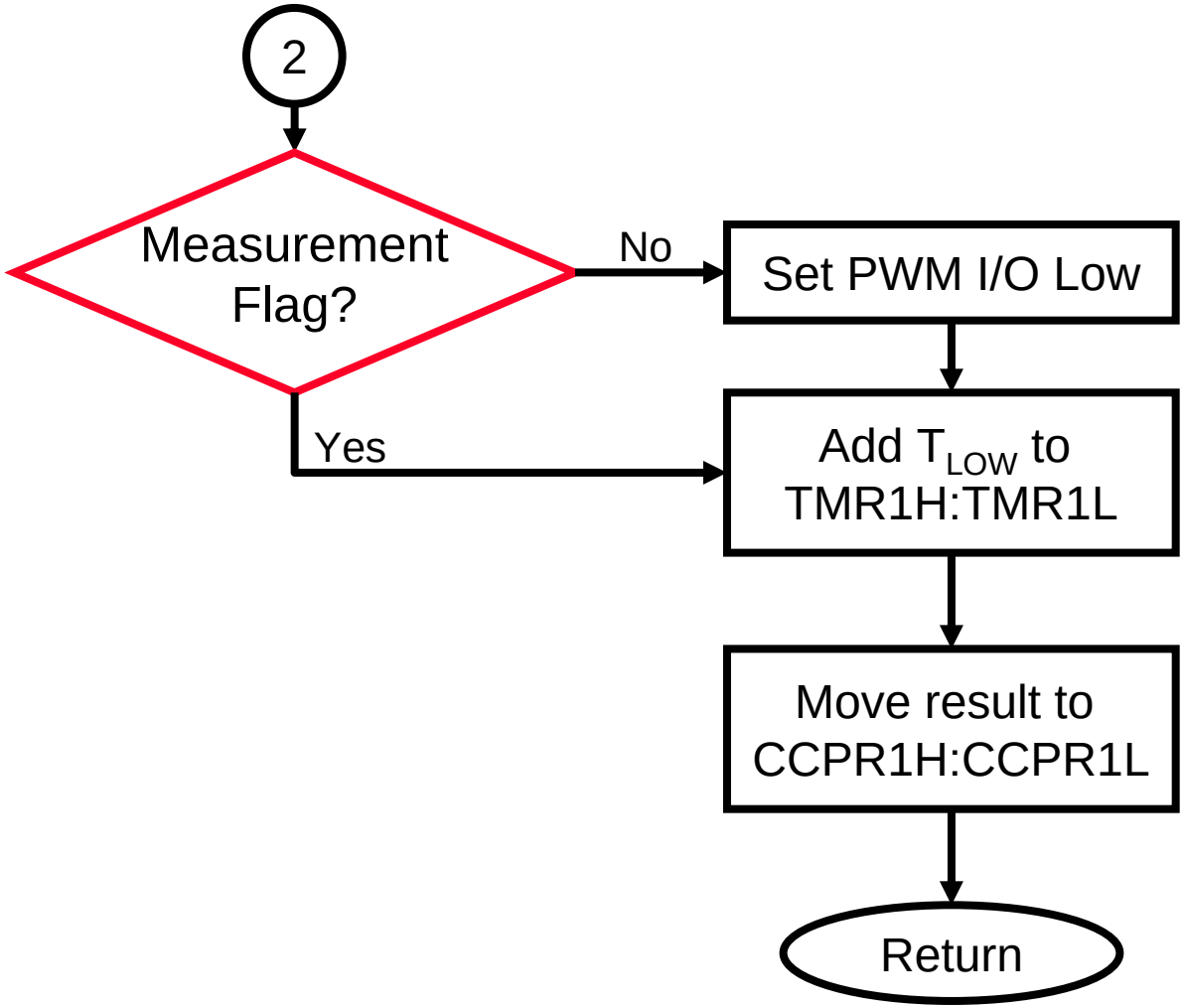
- Must leave the PWM line high if a measurement in progress
- Solution: Create a flag called **Measurement Flag** to indicate when a measurement is in progress

Setting PWM Duty Cycle



Software PWM Flowchart

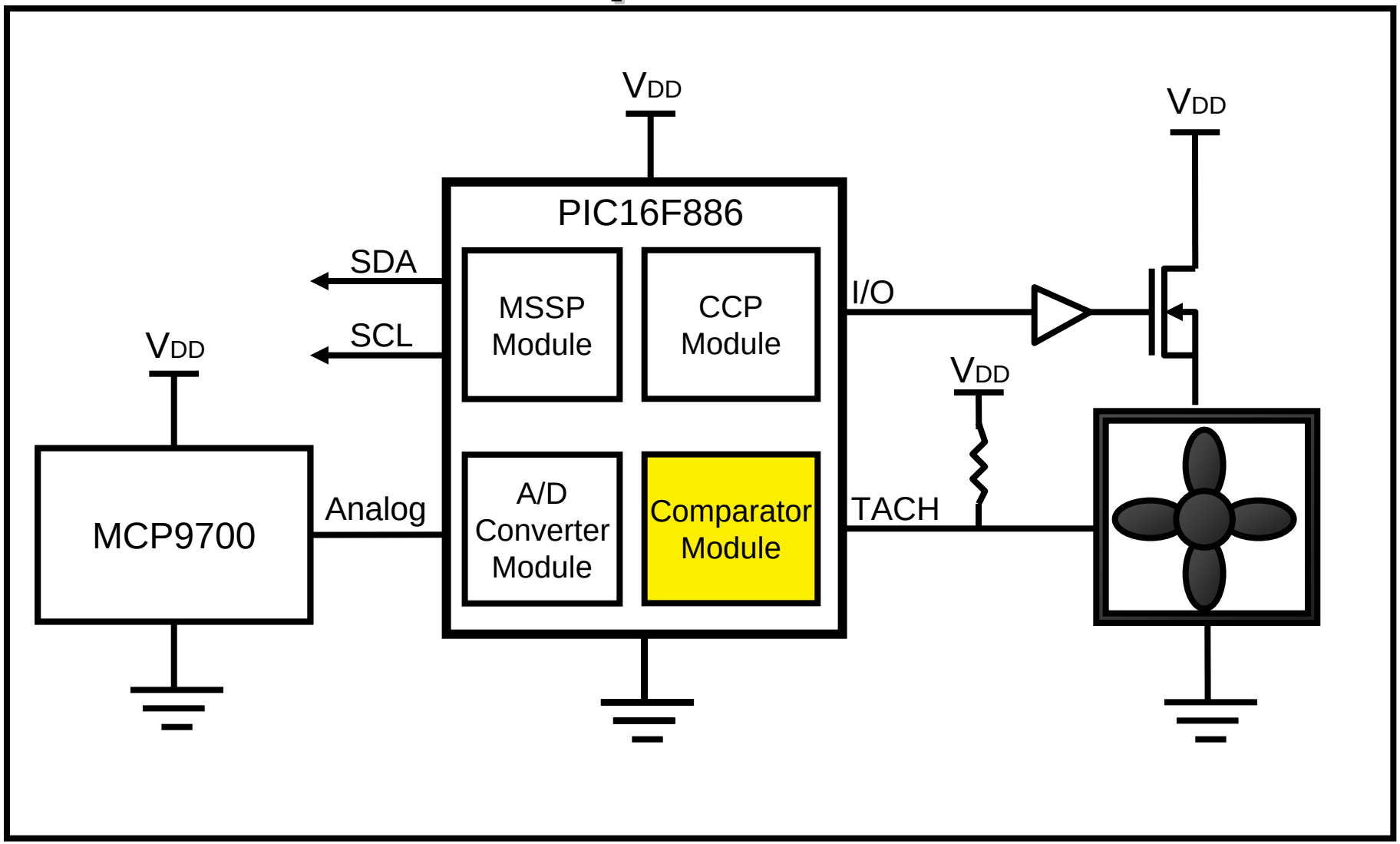
- CCP (Compare) Interrupt



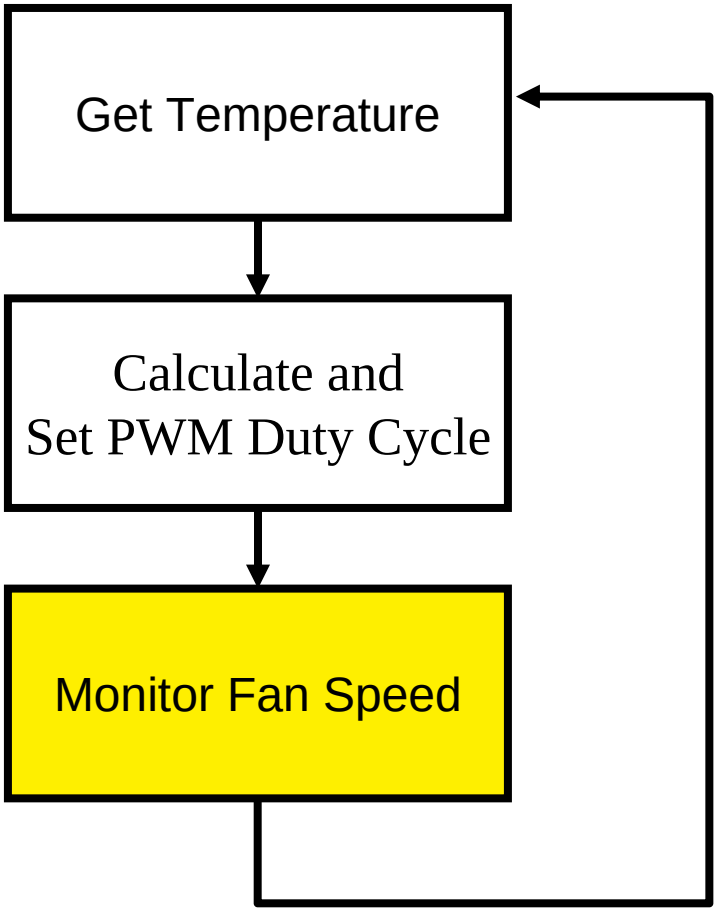
PWM Generation

Questions?

Thermal Controller Implementation



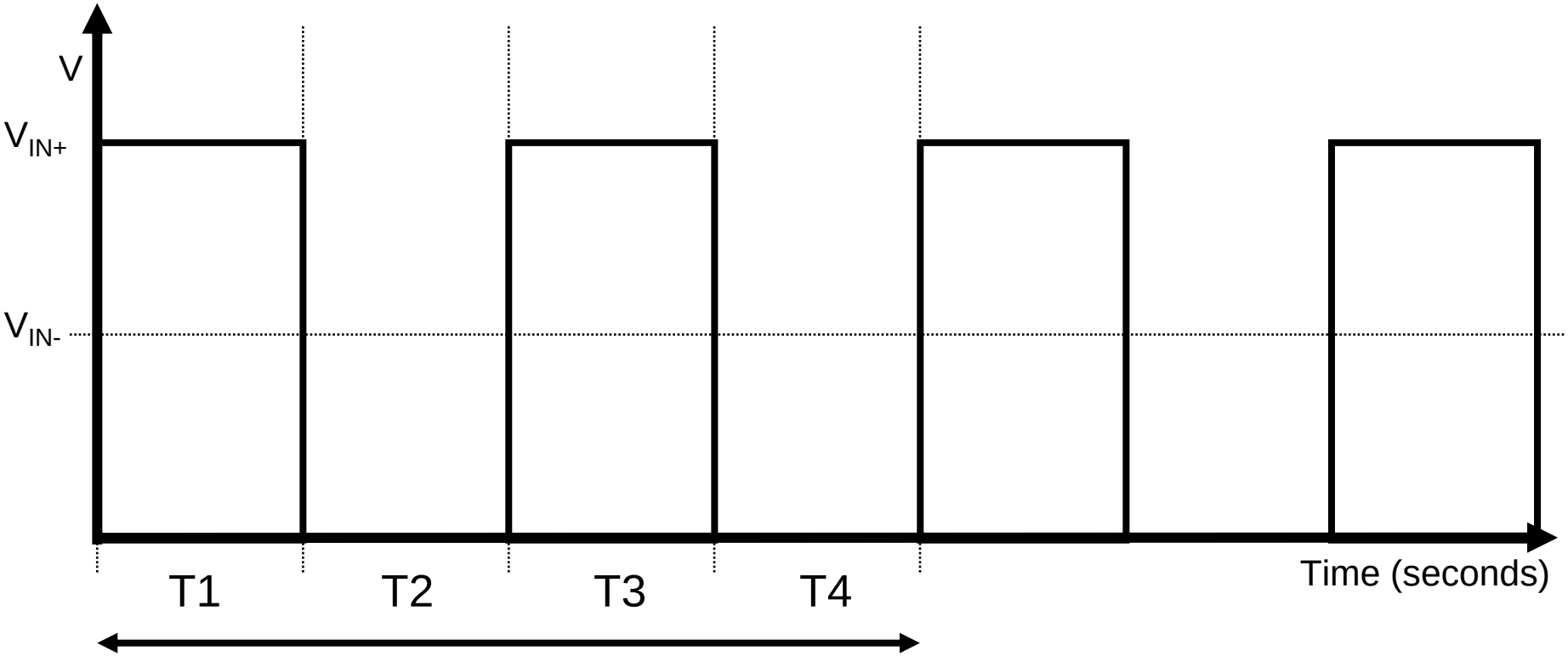
PIC® MCU Thermal Controller Flowchart



Measuring Fan Speed

- Comparator and timer 1 peripheral
- 4 transitions per revolution on TACH line
  - Fan datasheet specifies TACH line output

Tachometer Output



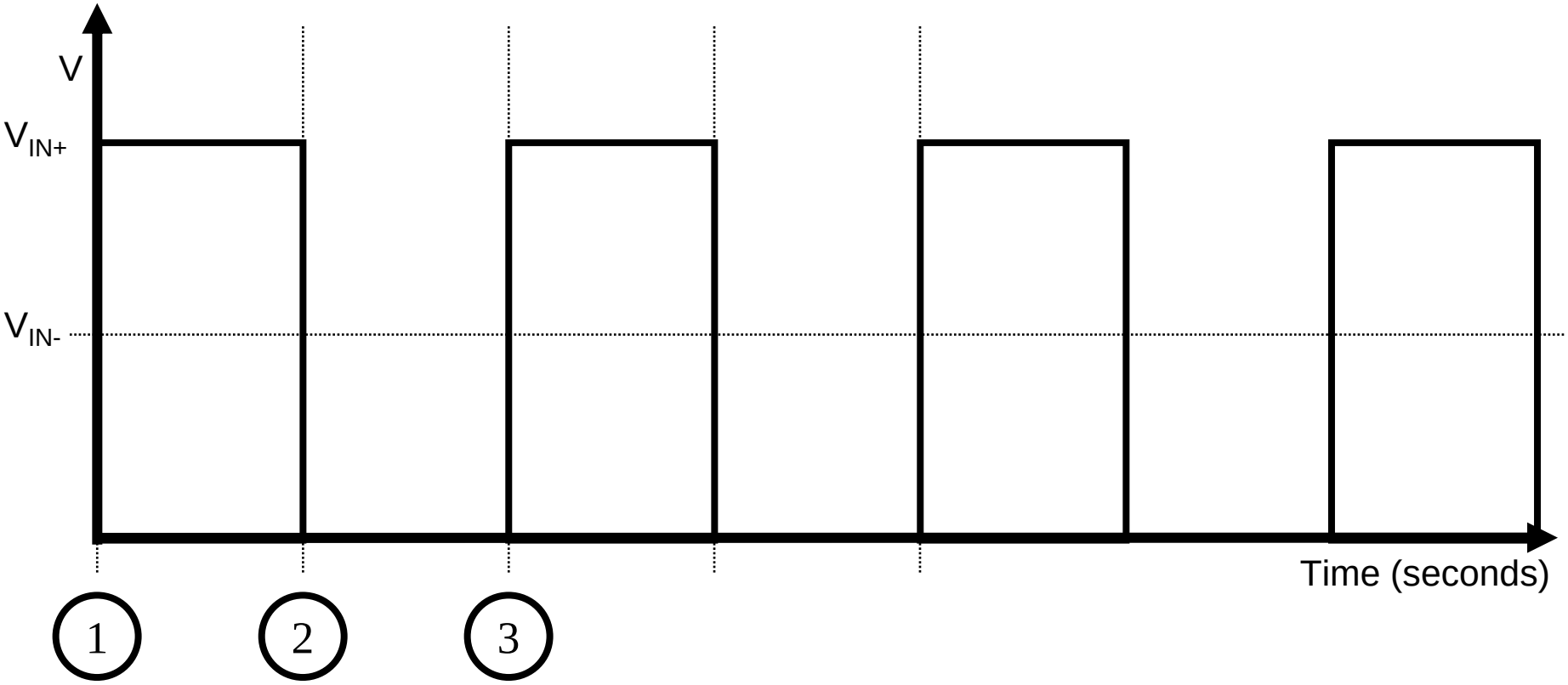
T = 1 Rotation

T = T1 + T2 + T3 + T4

T1 = T2 = T3 = T4 = 60 / (4 x RPM)

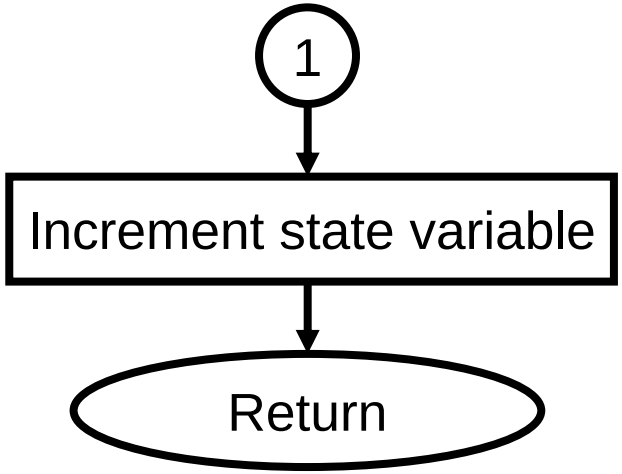
RPM = 60 / (4 x T1)

Comparator Interrupts

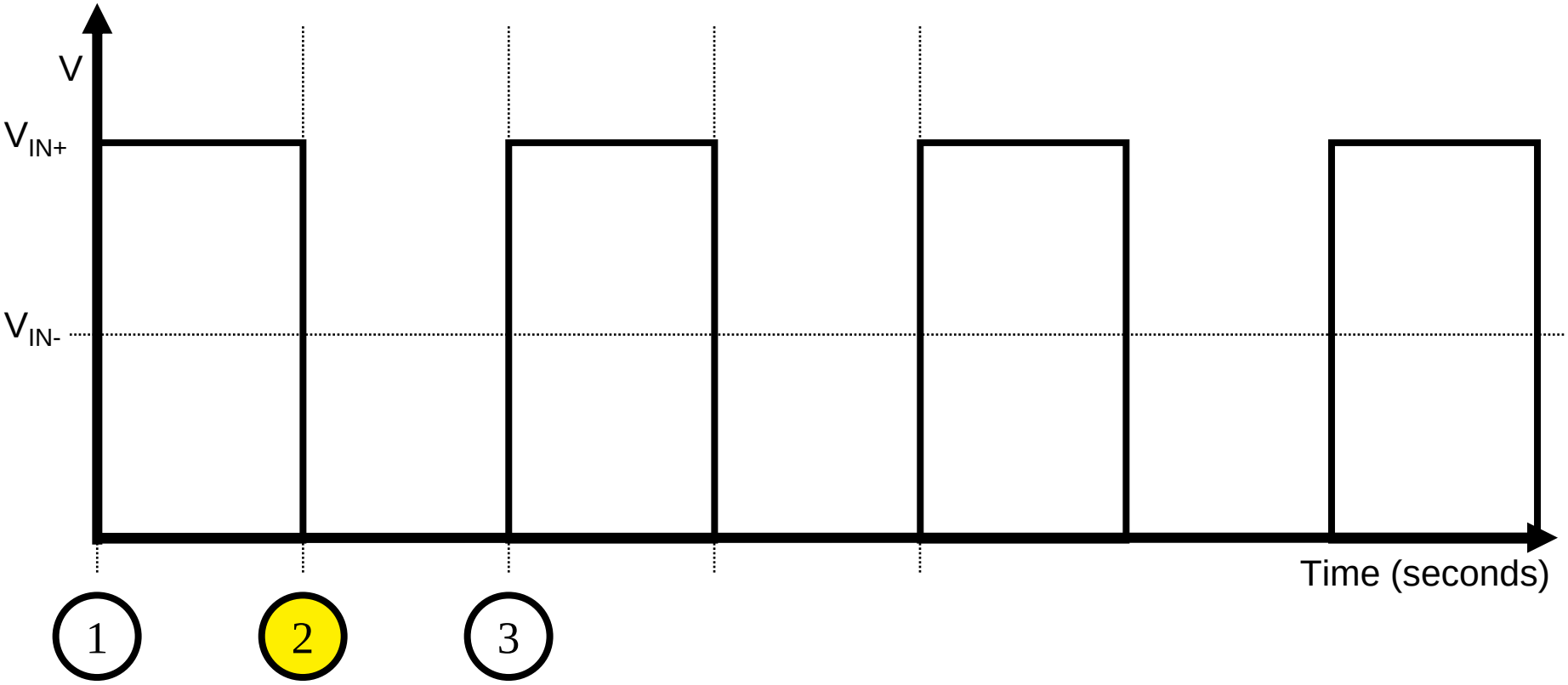


Measuring Fan Speed

- Comparator Interrupt
- Increment the state variable
- Ignore first edge (allows Hall sensor to stabilize)

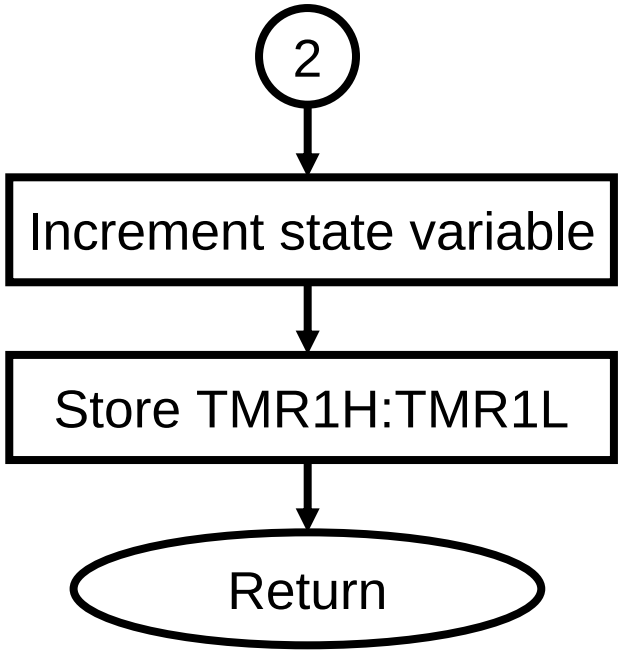


Comparator Interrupt 2

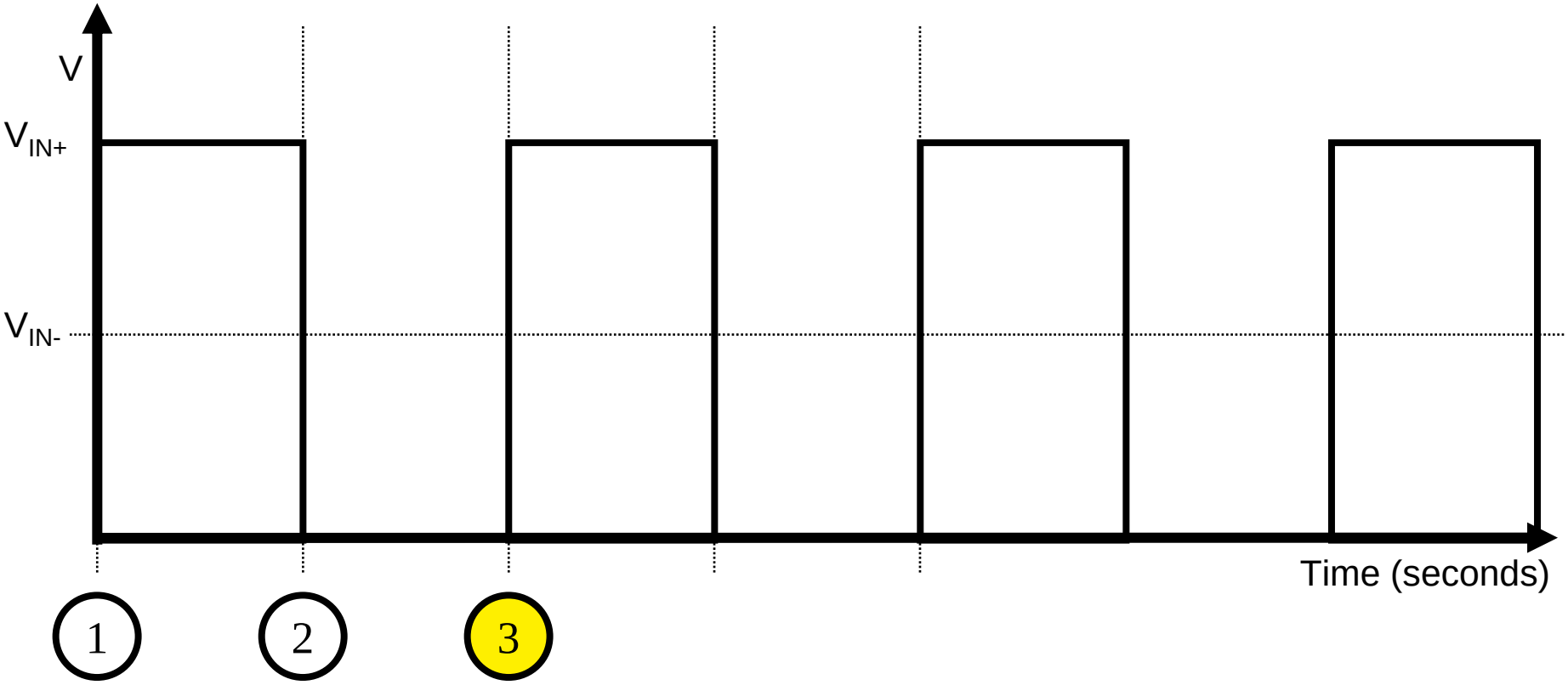


Measuring Fan Speed

- Comparator Interrupt
- Second interrupt stores Timer 1 value
- Begin Measurement

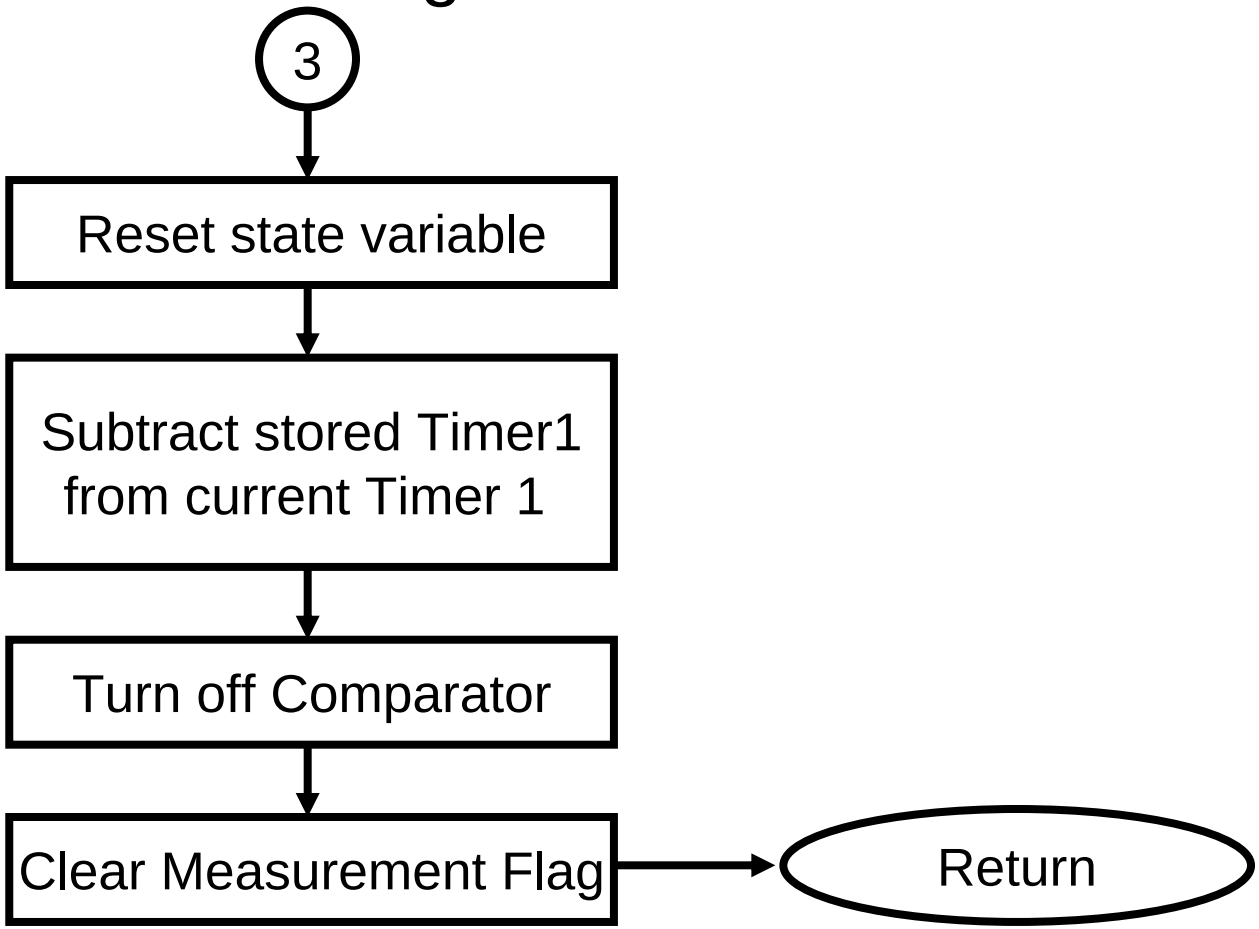


Comparator Interrupt 3

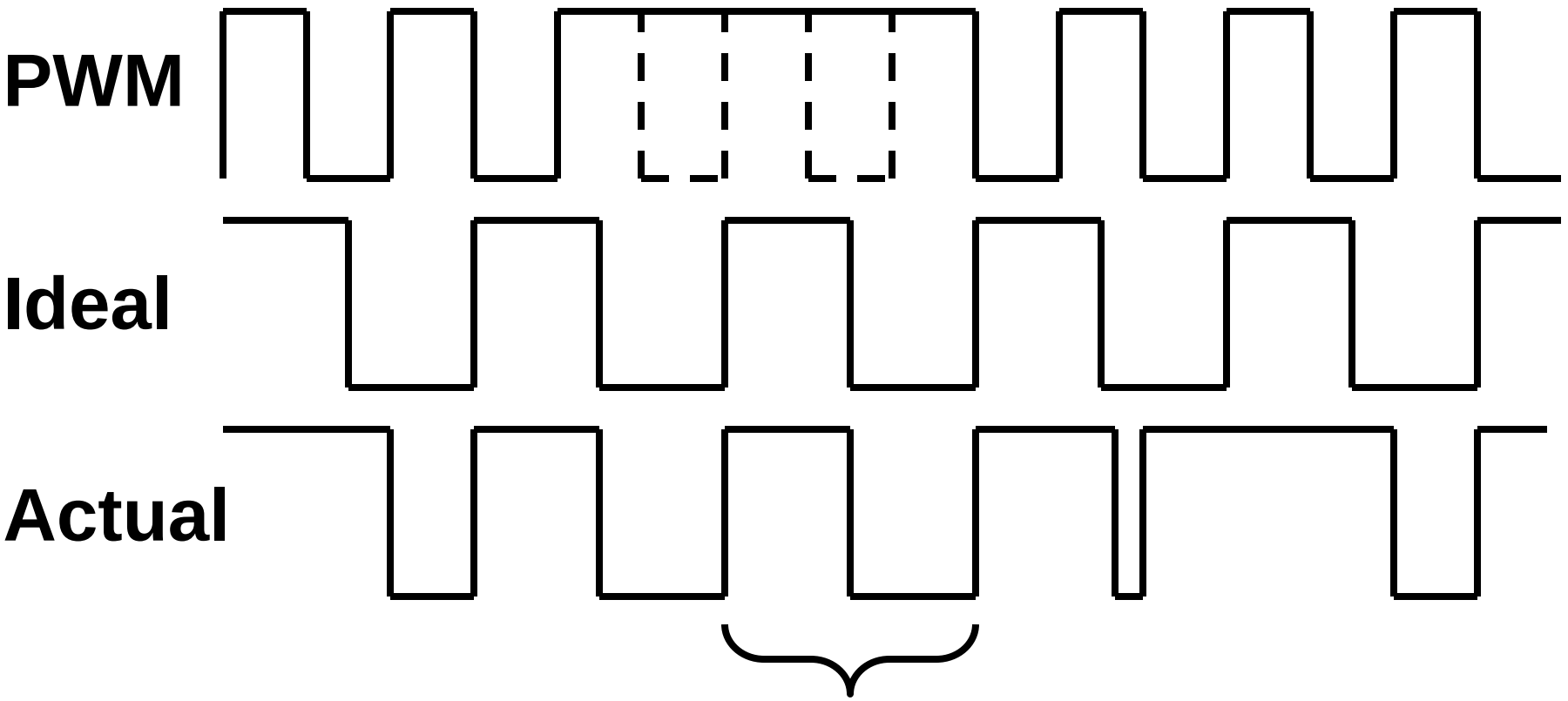


Measuring Fan Speed

- Comparator Interrupt
- Third determines the change in time

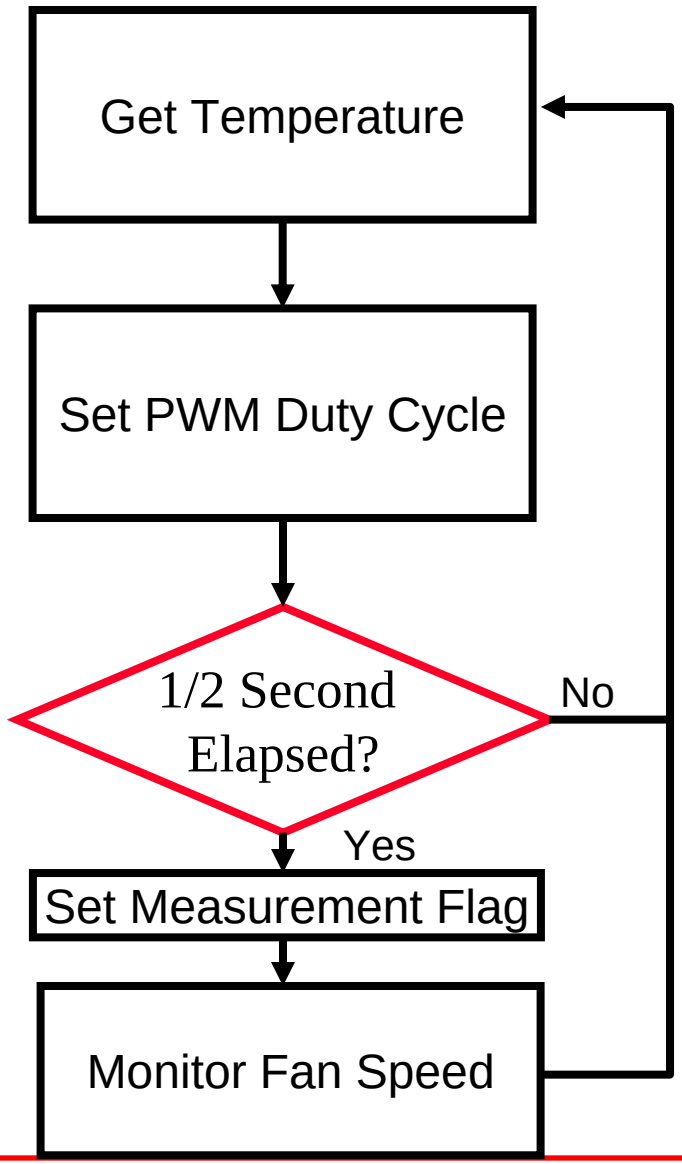


Problem: Measuring Too Often



- Fan will be full on during measurements
- Measuring too often can change the speed of a fan
- Not measuring enough may yield inaccurate speed measurements
- Solution: Measure every half second

Solution

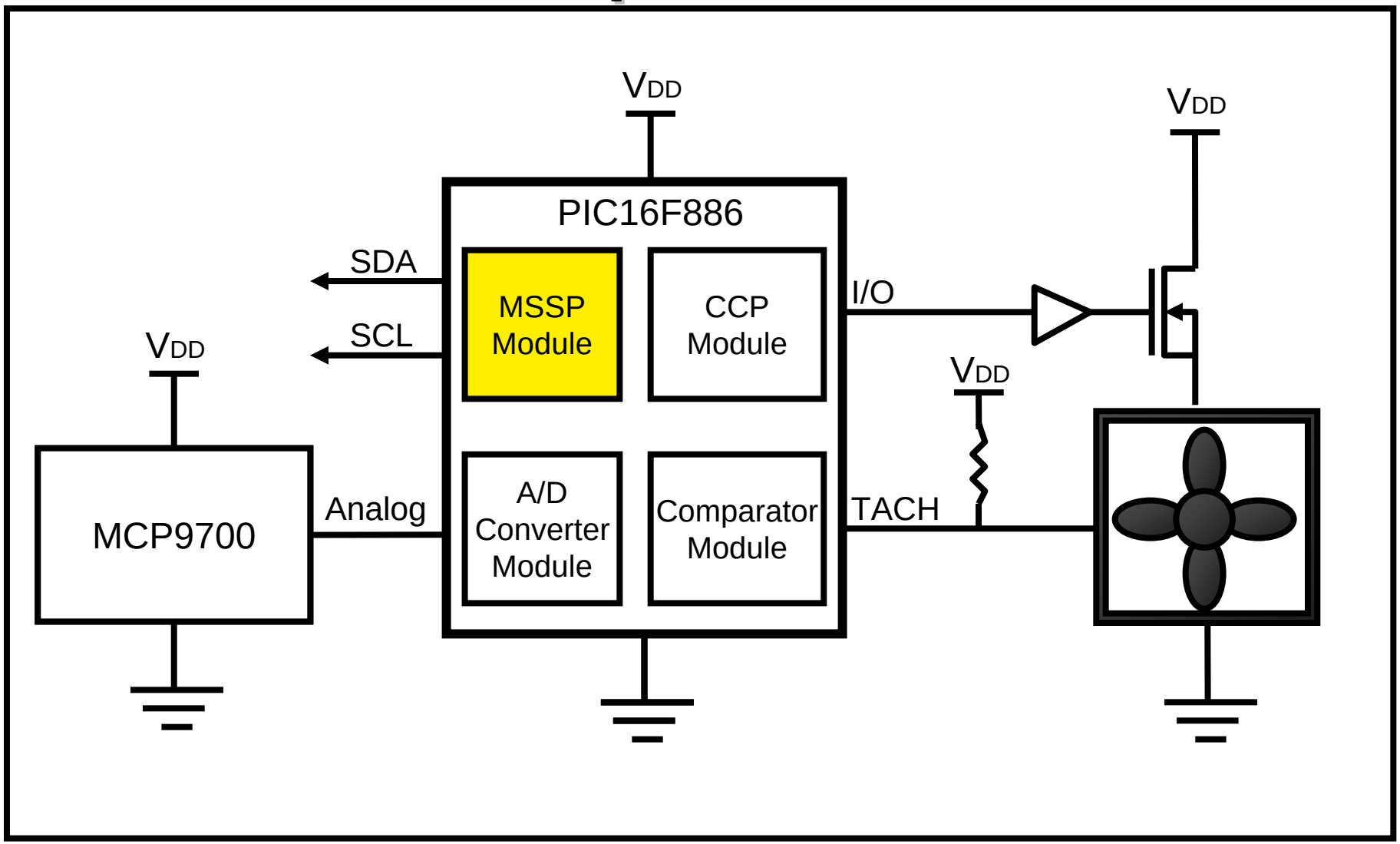


University of Microchip UofM MASTERs 2007

Measuring Fan Speed

Questions?

Thermal Controller Implementation

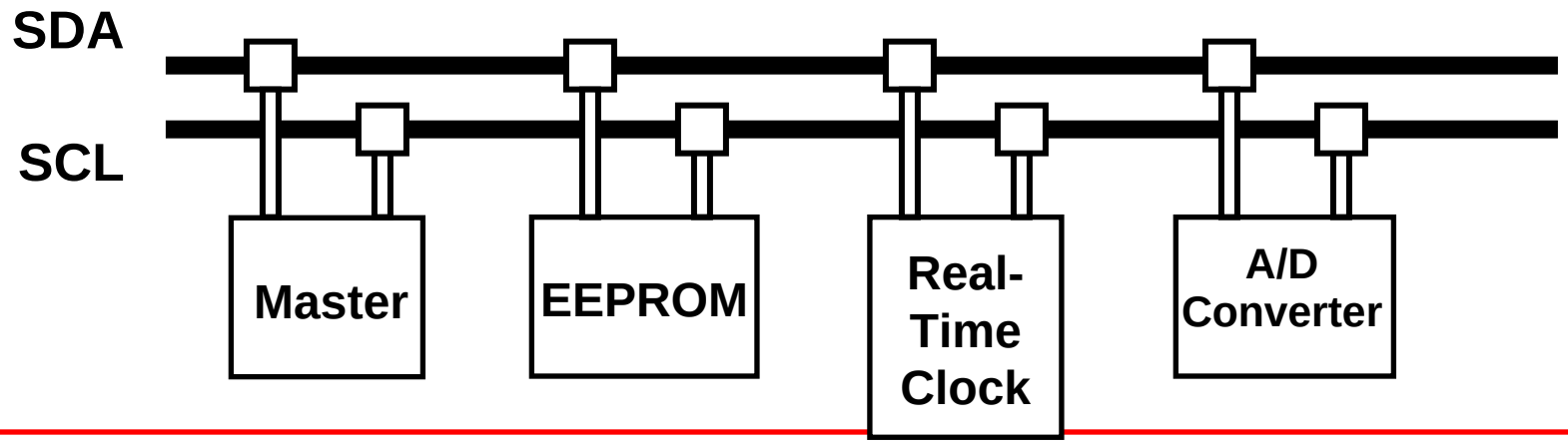


MSSP Basics

- MSSP I²C™ Mode Supports:
  - Master/Multi-Master Mode
  - Slave Mode
  - 7 or 10-bit Addressing
- Configurations:
  - I²C Master Mode
  - I²C Slave Mode
  - I²C Slave Mode with Start and Stop bit interrupts enabled
  - I²C firmware controlled master, slave is idle

I²C™ Basics

- NXP(Philips) Inter-Integrated Circuit (I²C)
  - Specification:
    www.standardics.nxp.com/literature/books/i2c/pdf/i2c.bus.specification.pdf
- Synchronous (not like RS232)
- Master-Slave Protocol
- Bidirectional
- Half Duplex Serial Interface



I²C™ Message Formatting

- Master **Write** to Slave

From master to slave

From slave to master

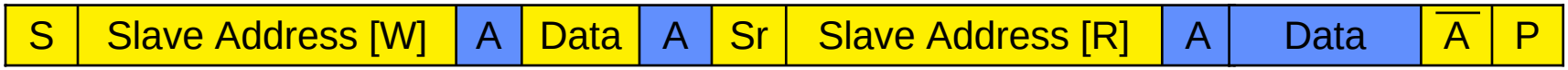
A = **Acknowledge**

S = **Start** Condition

P = **Stop** Condition

I²C™ Message Formatting

- Master **Combination Read** from Slave



From master to slave

From slave to master

A = **Acknowledge**

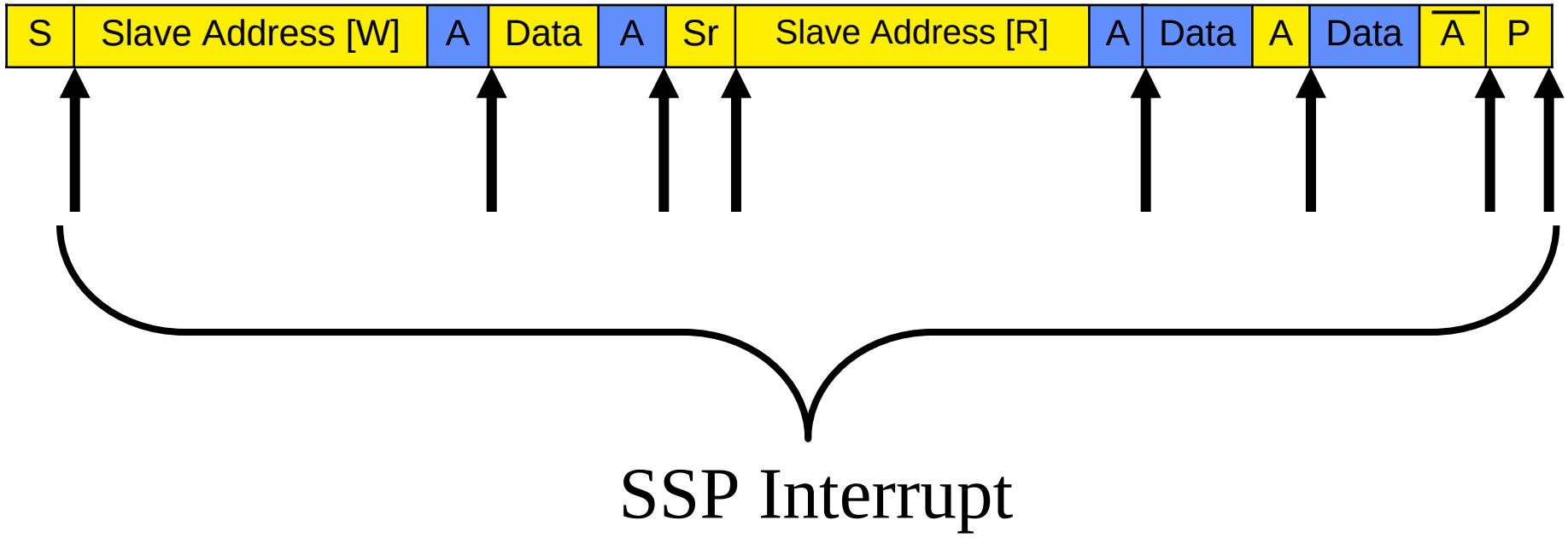
$\overline{A}$ = **No Acknowledge**

S = **Start**

Sr= **Repeated Start**

P = **Stop**

I²C™ Slave Mode Firmware Overview



Thermal Management Controller Registers

- Fan Controller writes data to GPR
- Fan Controller data is made read/writable through the I²C™ Slave Mode Firmware

General Purpose Registers

Offset Register	0x00
Temperature_H	0x01
Temperature_L	0x02
PWM High Time_H	0x03
PWM High Time_L	0x04
PWM High Time Max_H	0x05
PWM High Time Max_L	0x06
PWM High Time Min_H	0x07
PWM High Time Min_L	0x08
PWM Period_H	0x09
PWM Period_L	0x0A

Temperature Max_H	0x0B
Temperature Max_L	0x0C
Temperature Min_H	0x0D
Temperature Min_L	0x0E
Fan Speed_H	0x0F
Fan Speed_L	0x10
Fan Status	0x11

Summary

- Thermal Management Basics
- 3-Wire Fans
- Thermal Controllers
- Implementing a Thermal Controller on a PIC16F886

UNIVERSITY OF MICROCHIP UOFM MASTERS 2007

Trademarks

The Microchip name and logo, the Microchip logo, Accuron, dsPIC, KeeLoq, KeeLoq logo, microID, MPLAB, PIC, PICmicro, PICSTART, PRO MATE, rfPIC and SmartShunt are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

AmpLab, FilterLab, Linear Active Thermistor, Migratable Memory, MXDEV, MXLAB, SEEVAL, SmartSensor and The Embedded Control Solutions Company are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Analog-for-the-Digital Age, Application Maestro, CodeGuard, dsPICDEM, dsPICDEM.net, dsPICworks, ECAN, ECONOMONITOR, FanSense, FlexROM, fuzzyLAB, In-Circuit Serial Programming, ICSP, ICEPIC, Mindi, MiWi, MPASM, MPLAB Certified logo, MPLIB, MPLINK, PICkit, PICDEM, PICDEM.net, PICLAB, PICtail, PowerCal, PowerInfo, PowerMate, PowerTool, REAL ICE, rfLAB, Select Mode, Smart Serial, SmartTel, Total Endurance, UNI/O, WiperLock and ZENA are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.

All other trademarks mentioned herein are property of their respective companies.