UNIVERSITY OF MICROCHIP
UofM
MASTERs 2007

11036 I2C

Using an I²C™ EEPROM to Implement a Temperature Datalogger

University of Microchip UofM MASTERs 2007

Class Objectives

When you walk out, you will know:

- **I²C™ Bus Basics.**
  - Start, Stop, ACK, Bus Transfer operations.
- **How to implement High Level I²C in software on PIC10F family**
- **How to address and communicate with more than one device on the Bus**
- **How to integrate this into a complete design, which you will keep…**

UNIVERSITY OF MICROCHIP UOFM MASTERS 2007

Class Agenda

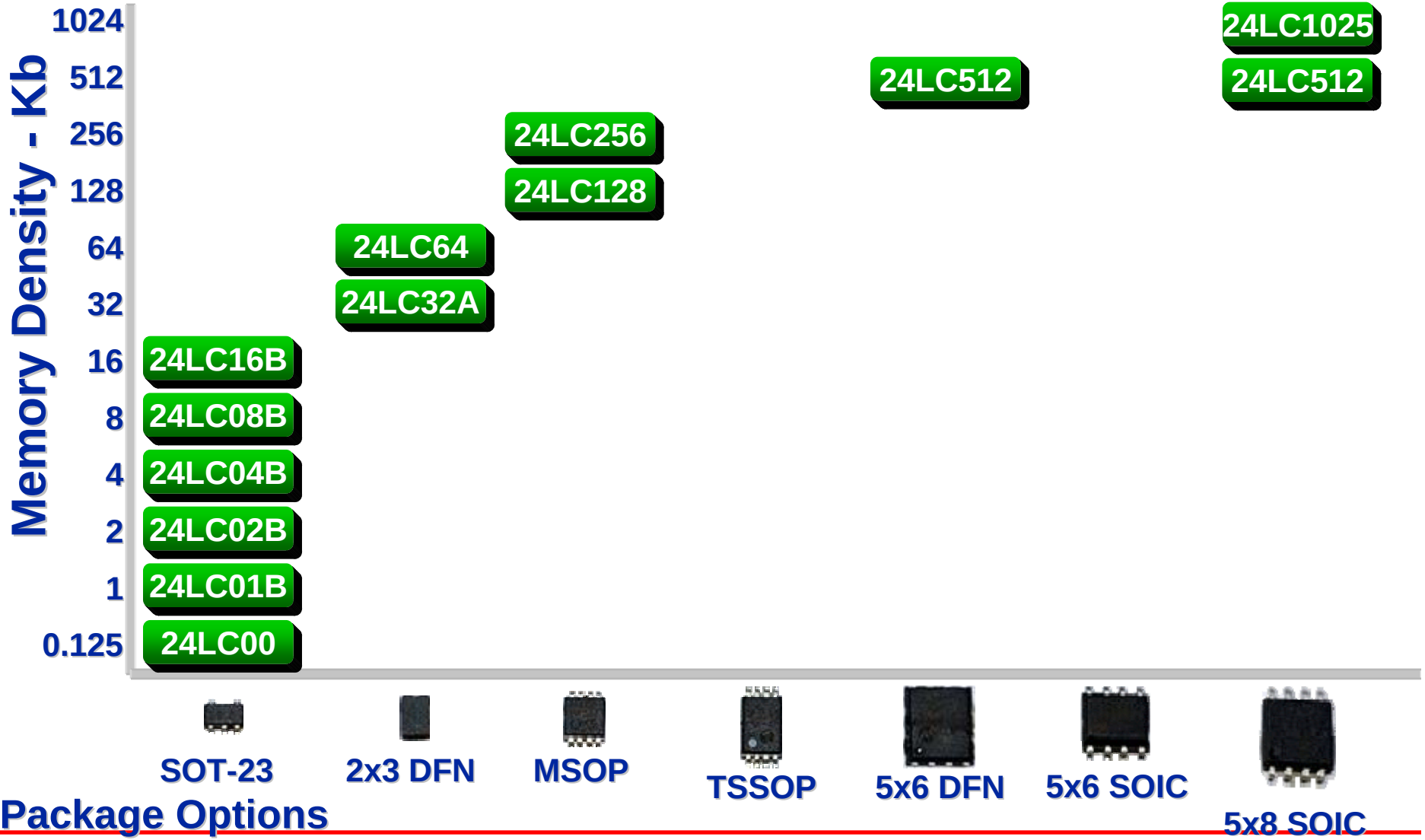
- **Update on Microchip I²C™ Devices**
- **I²C Bus Specification**
  - Start Condition
  - Stop Condition
  - ACK/NACK Conditions
  - Bus Data Transfer
- **You will then use the information to complete the code of a temperature logger and test your application**

University of Microchip UofM MASTERs 2007

I²C™ Products from Microchip Technology

University of Microchip UofM MASTERs 2007

Microchip I²C™ Serial EEPROMs



The I²C™ Bus Specification

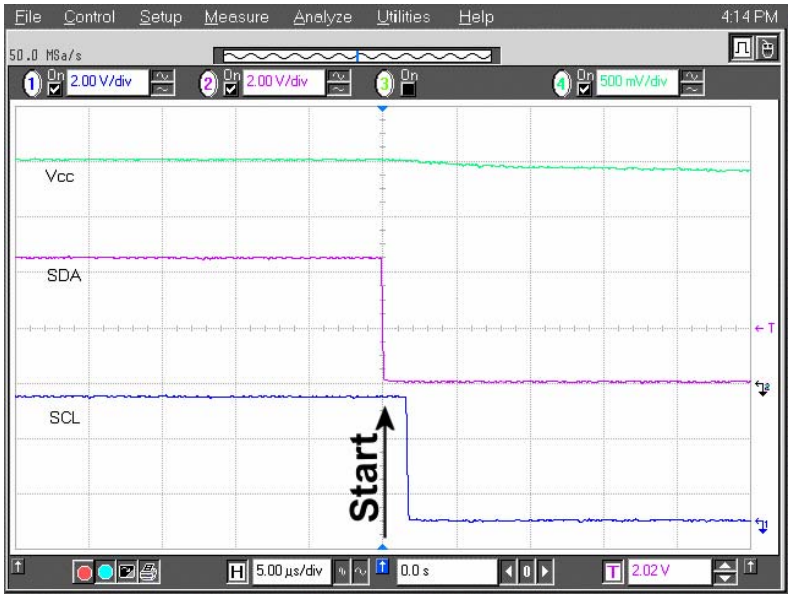
I²C™ Bus Specification

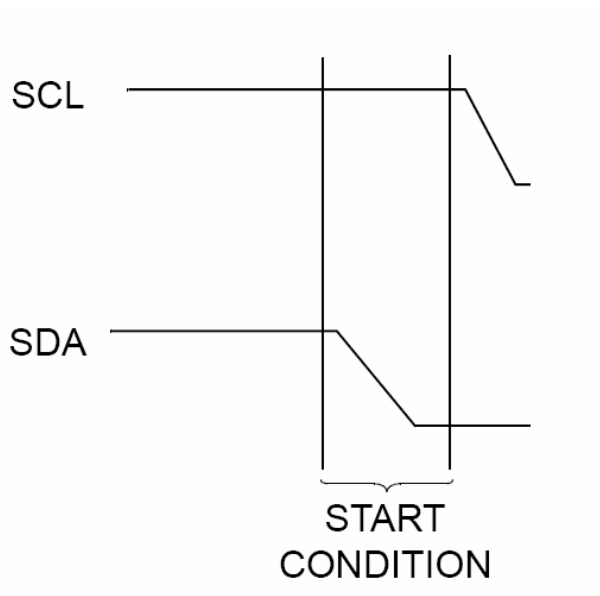
- **Developed by Philips**
  - Addressable 2-wire bus capable of addressing 8 serial EEPROM devices in addition to other devices, such as Temperature Sensors
  - Defined Start and Stop Conditions
  - Defined Bus Arbitration
  - Synchronous Master/Slave system
  - 100 kHz/400 kHz and 1 MHz bus speeds
  - SDA – Serial Data
  - SCL – Clock Line
  - Both SDA and SCL require pull-up resistors

I²C™ Start Condition

I²C™ Start Condition

- **Used to Initiate the Start of a period of Bus Activity**
  - Defined as a High to Low Transition of SDA while the SCL line is Held High

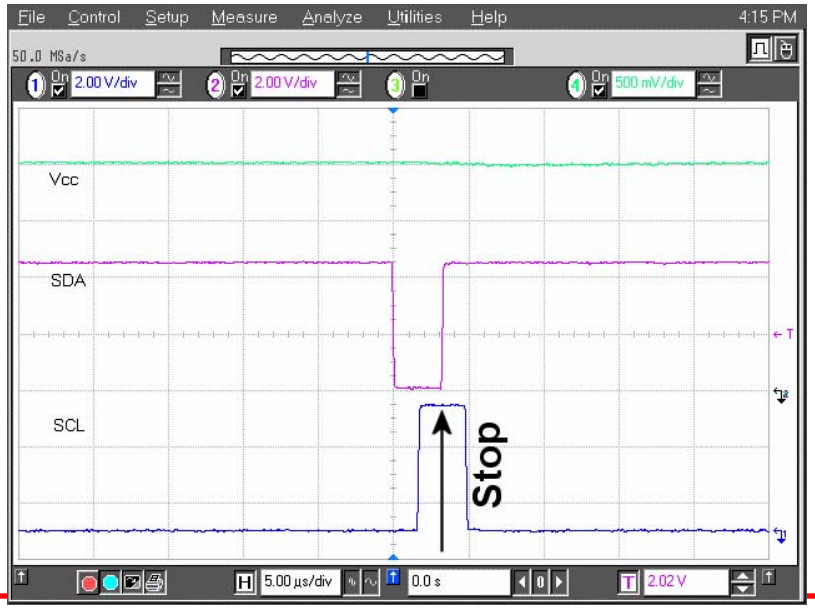


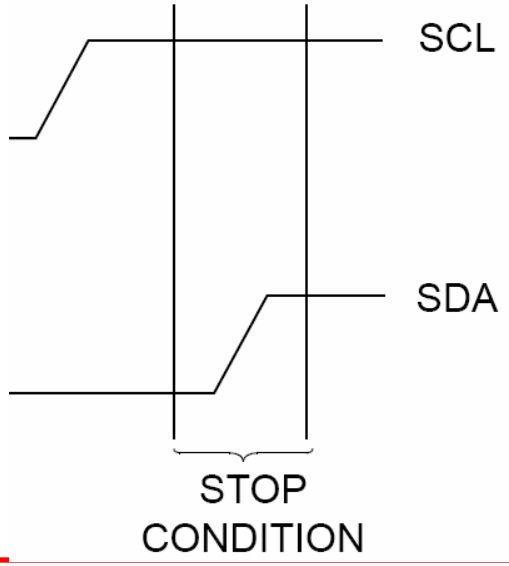


I²C™ Stop Condition

I²C™ Stop Condition

- **Used to Signal the End of a period of Bus Activity**
  - Defined as a Low to High Transition of SDA while the SCL line is Held High

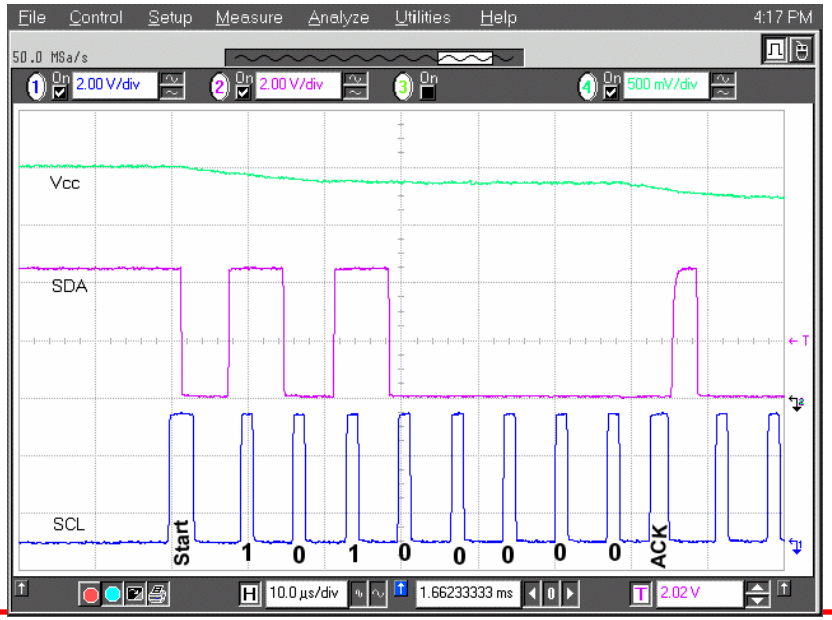


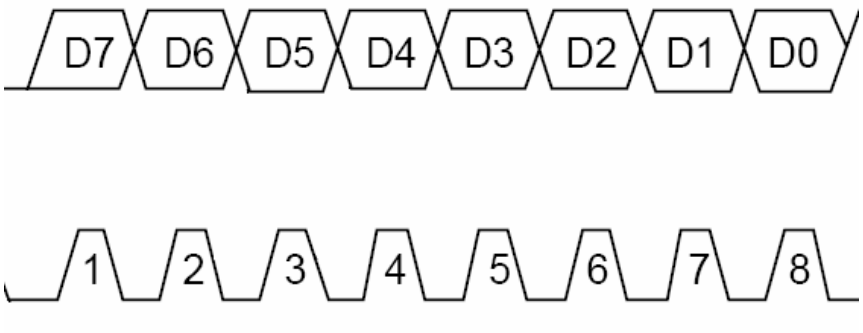


I²C™ Data Transfer

I²C™ Data Transfer

- **Data is transferred from the Master to the Slave in blocks of 8 bits**
  - Data is transferred on the rising edge of SCL

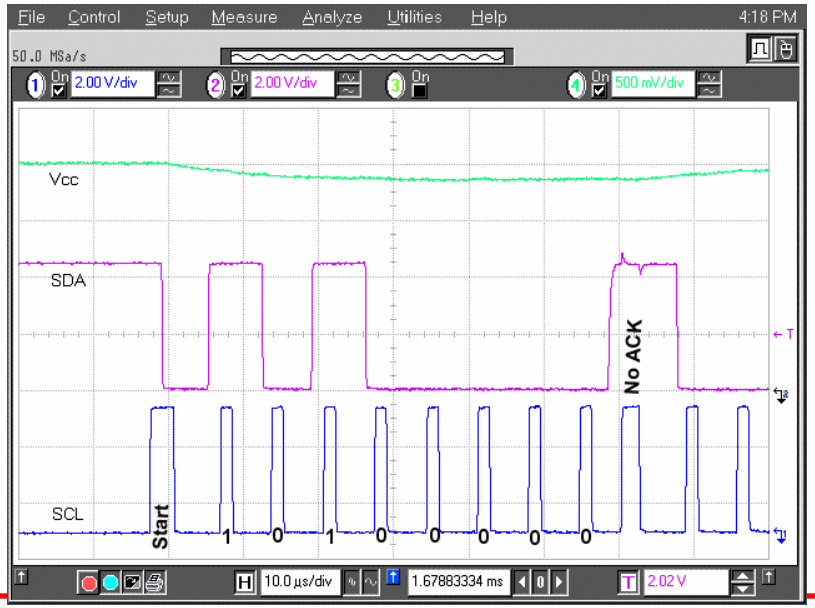


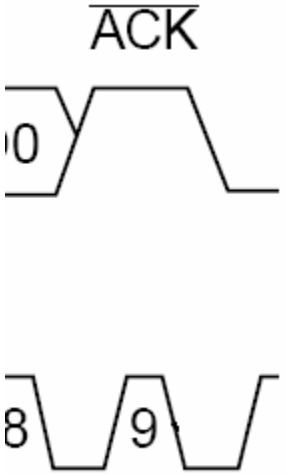


I²C™ ACK/NACK Condition

I²C™ ACK/NACK Condition

- **Used to indicate the Success or failure of a data transmission or continuation of an operation**
  - Generated by the Master or the Slave by holding SDA on the 9th rising SCL

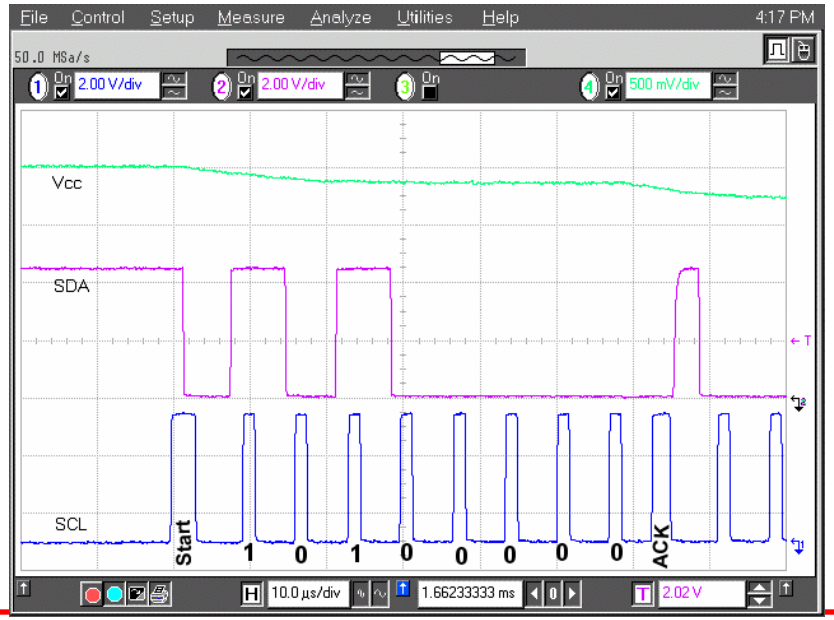


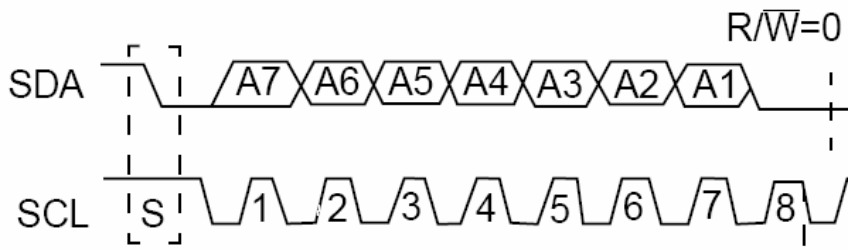


I²C™ Device Addressing

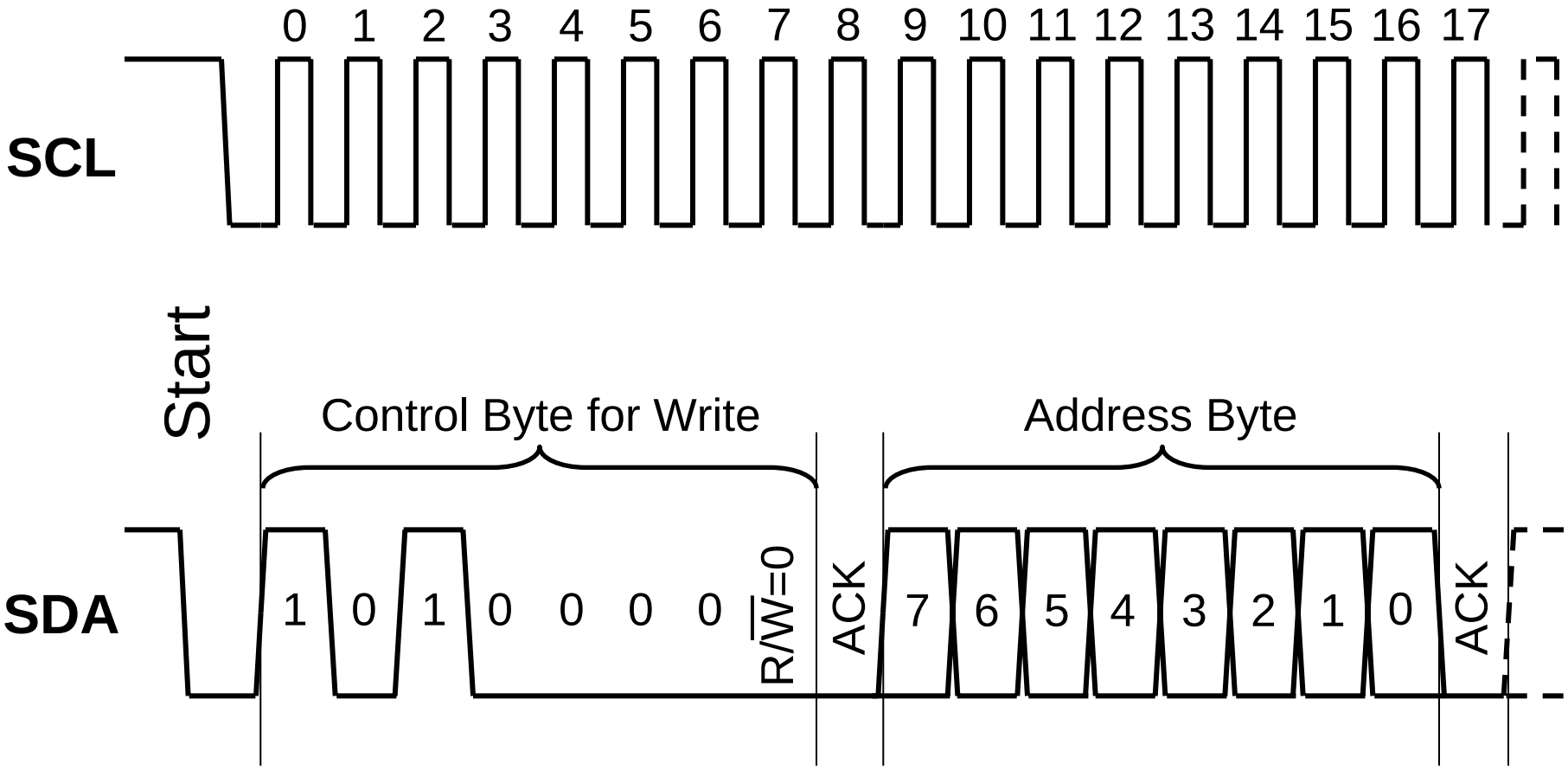
I²C™ Device Addressing

- **The first byte received following the Start condition is the control byte**
  - It contains the control code, block- or chip-select bits, and the R/W bit

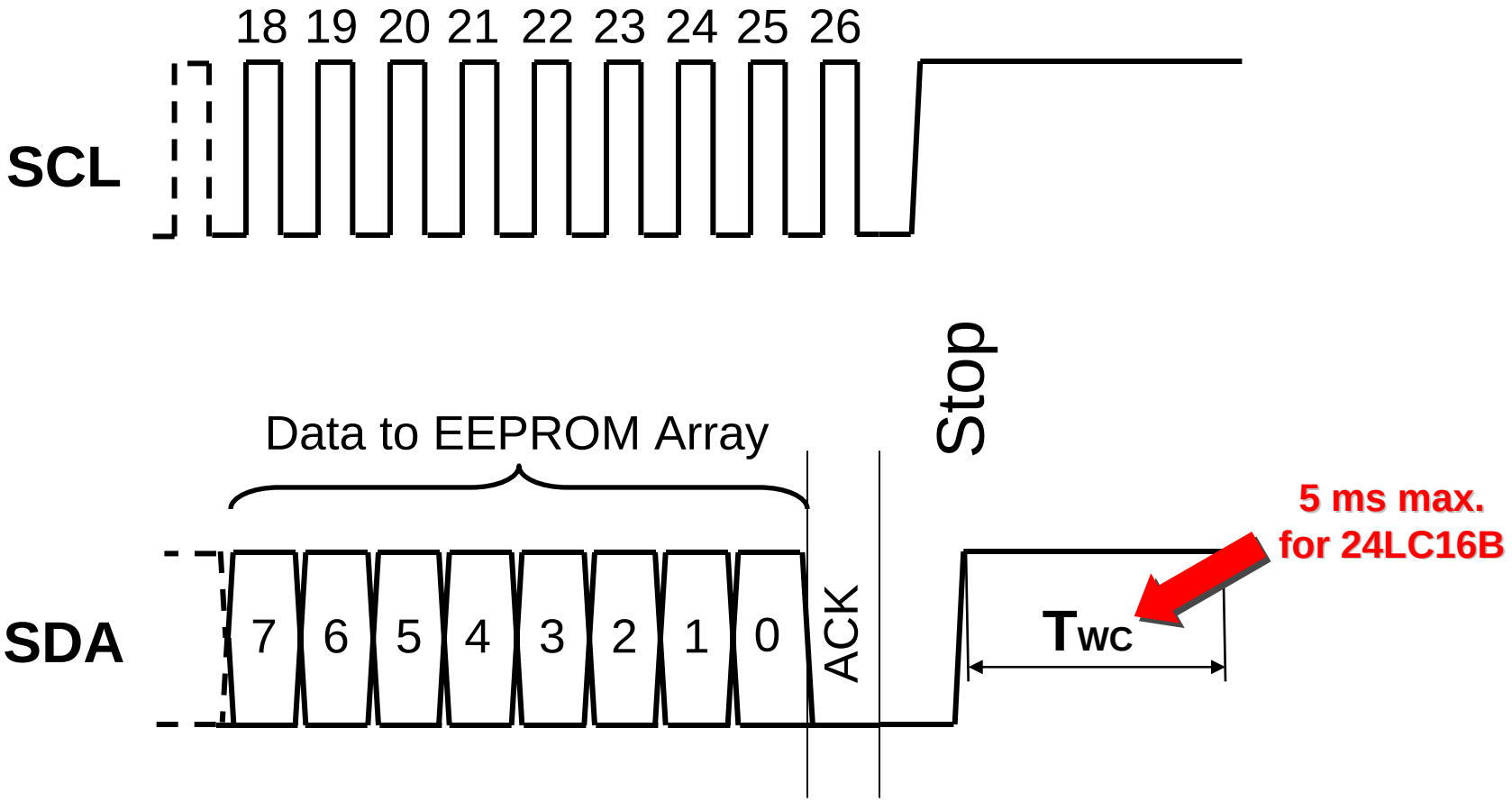




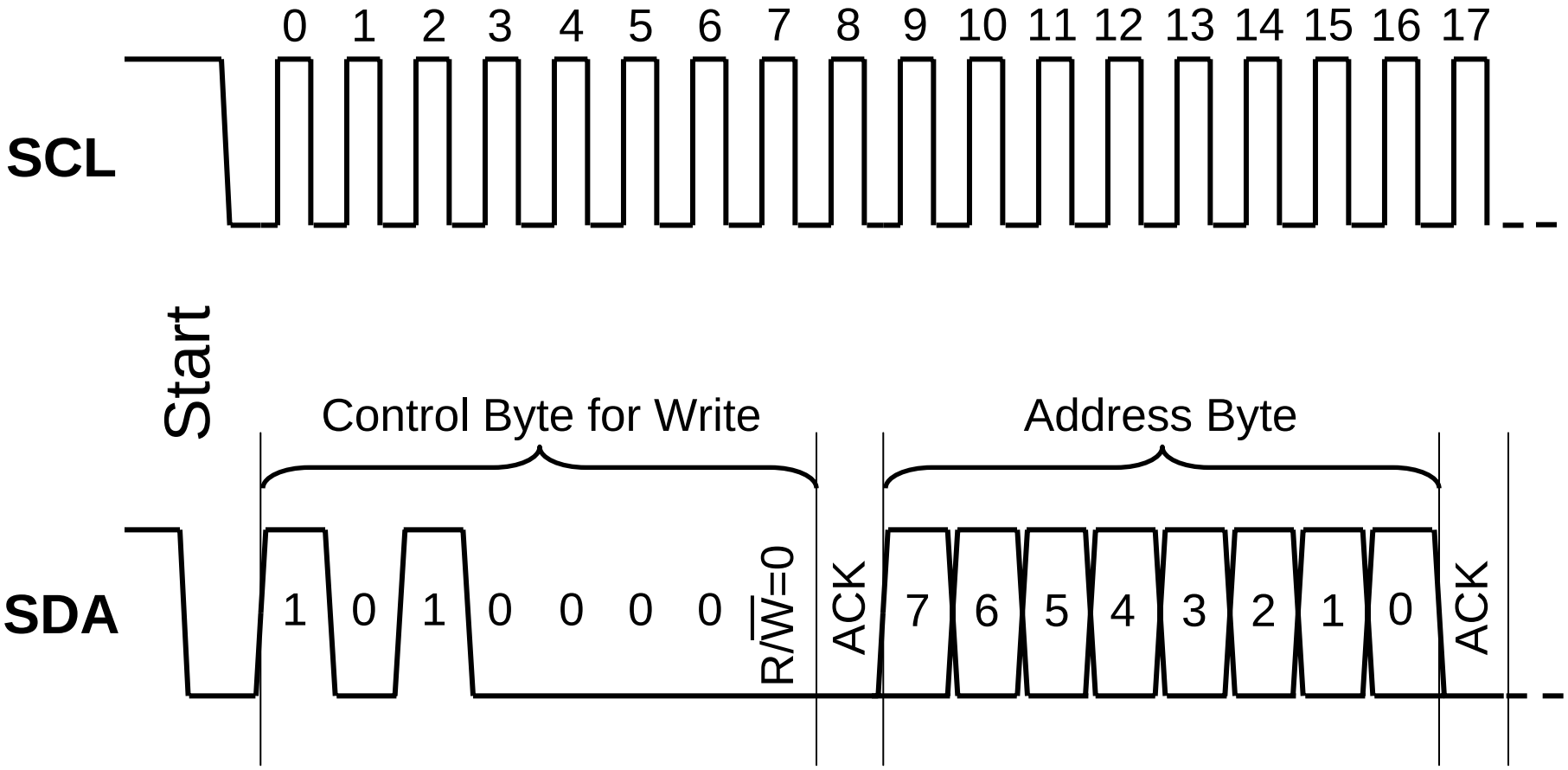
I²C™ Example Write Operation



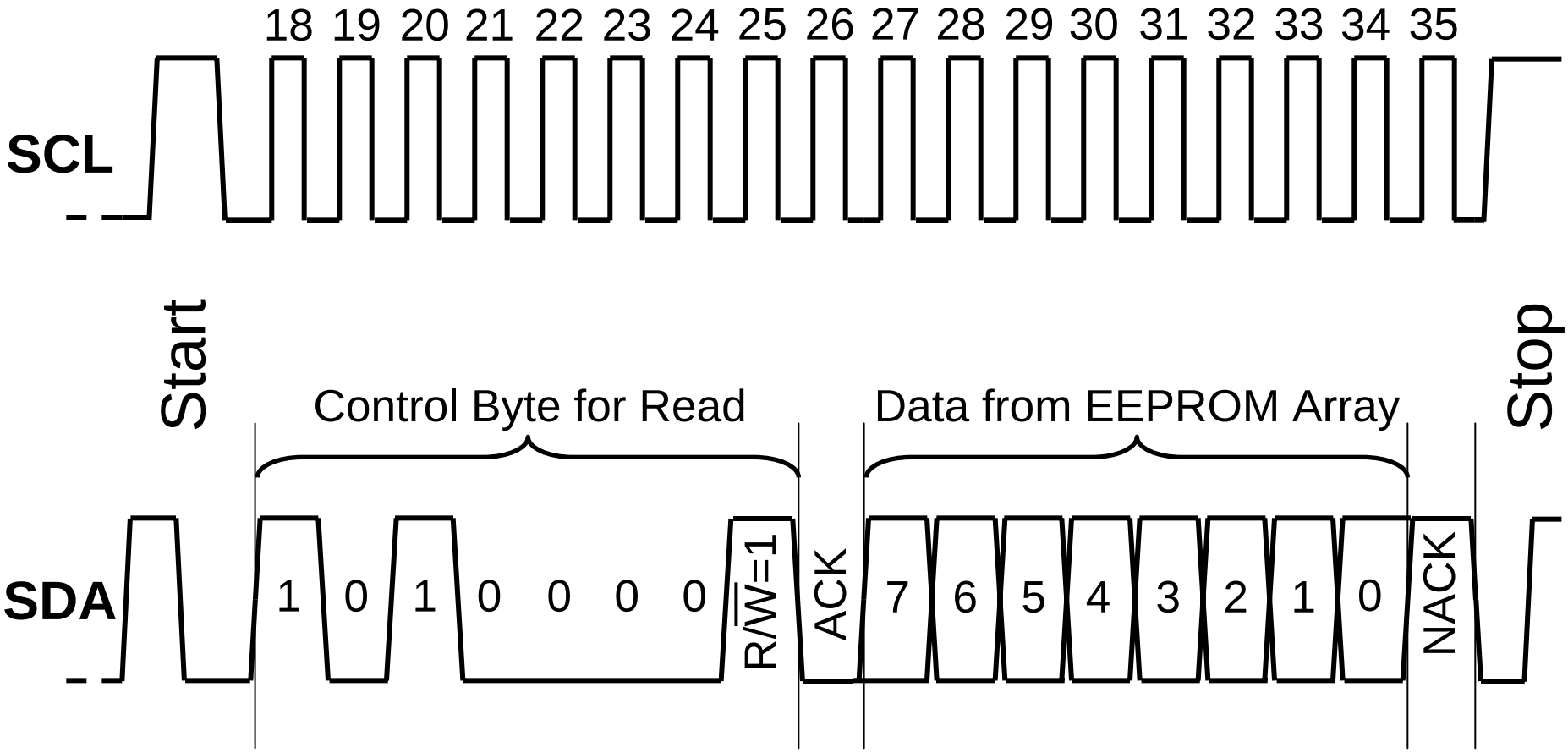
I²C™ Example Write Operation (cont'd.)



I²C™ Example Read Operation



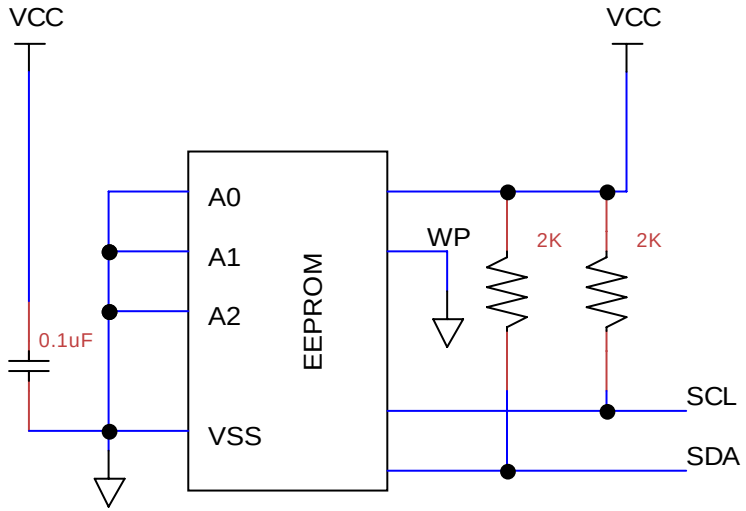
I²C™ Example Read Operation (cont'd.)

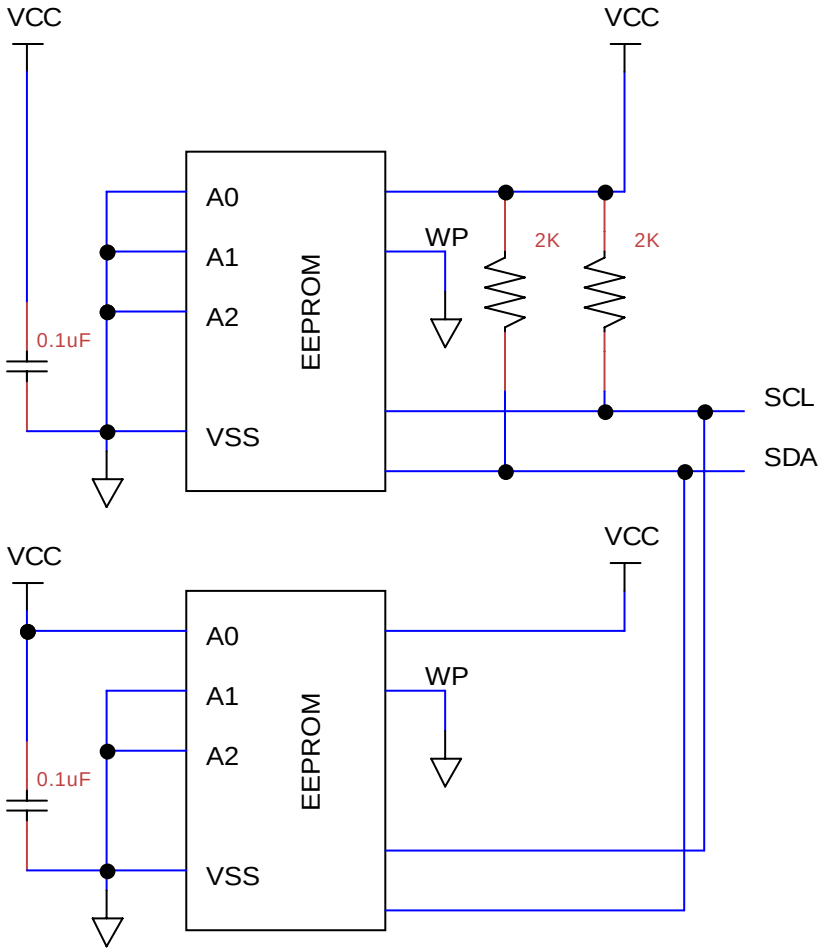


I²C™ Summary

- **Two Wire Bus**
  - DC – 100/400 kHz bus speed up to 1 MHz extended bus speed
  - Master/Slave configuration
- **Start Condition**
  - SDA – High to Low with SCL held High
- **Stop Condition**
  - SDA – Low to High with SCL held High
- **ACK/NACK Condition**
  - SDA – held by Master/Slave on 9th Rising SCL

I²C™ Tricks and Tips - 1





I²C™ Tricks and Tips - 2

- **Pull up Resistor on SCL ensures power up into standby (and is required for multimaster)**
- **Address pins can be any combination**
  - All ‘0’ is easiest
- **WP=0 allows for writes to occur**
  - WP high makes a serial ROM
- **Pullup on SDA needs to be set for bus speed**
  - Suggest 10KΩ for 100 kHz or less
  - Suggest 2KΩ for higher speed (or less if bus capacitance is high) – Review AN1028 for information on how to Calculate Values

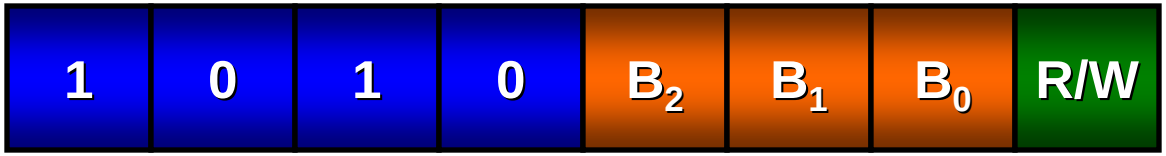
I²C™ Tricks and Tips - 3

- **Save time in your application**
  - I²C functions are available for all micros from PIC10F through to dsPIC33 family
  - Functions for reading and writing to Serial EEPROMS
  - Variants for micros with hardware MSSP module and micros without

Devices Used in This Class

Microchip 24LC16B Serial EEPROM

- 16 Kb (2 KB) Array Size
- 16-Byte Page Write Buffer
- 400 kHz Max. Clock Frequency
- Control Byte Allocation:



  - Control Code
  - Block-Select Bits
  - Read/Write Bit

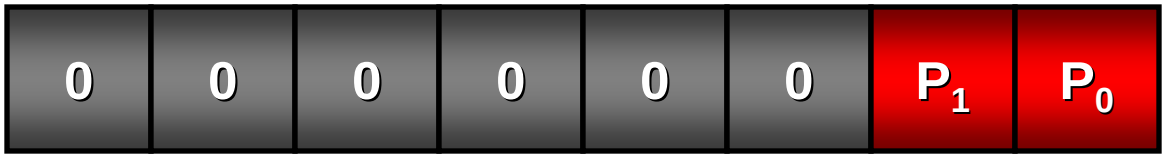
Microchip MCP9800 Temperature Sensor

- **User-Selectable 9 – 12-Bit Resolution**
  - Firmware uses 9-bit resolution
- **30 ms Typical Conversion Time (9-bit Resolution)**
- **Control Byte Allocation:**

  - Control Code
  - Slave Address Bits (class board uses '000')
  - Read/Write Bit

MCP9800 Operation

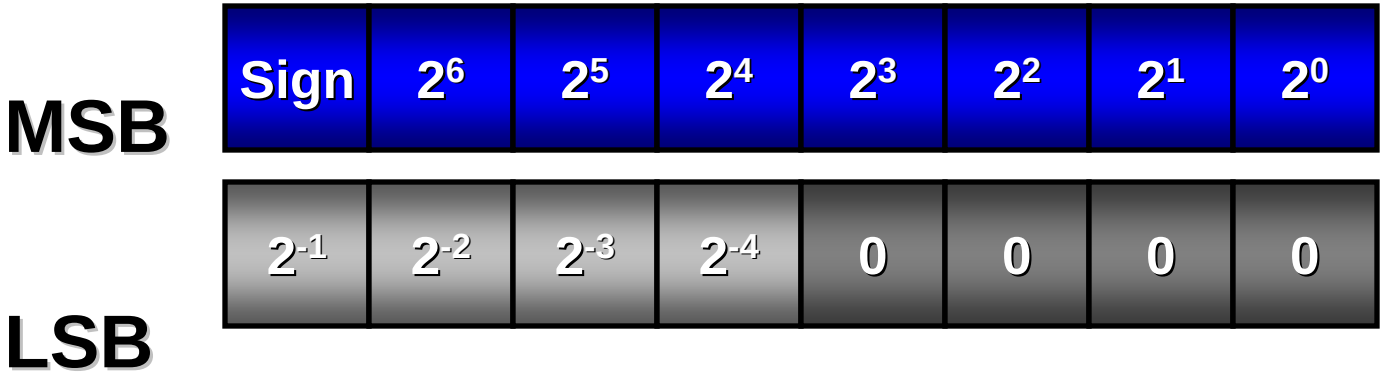
- **Register Pointer**



  - Unimplemented
  - Pointer Bits
    - **’00’ = Temperature Register**
    - **’01’ = Configuration Register**
    - **’10’ = Temp. Hysteresis Register (not used)**
    - **’11’ = Temp. Limit-Set Register (not used)**
- **The first data byte sent after the control code sets the register pointer**

MCP9800 Operation (cont'd.)

- **Temperature Register**



  - Unimplemented
  - Temperature Value (LSB not used in this class)

MCP9800 Operation (cont'd.)

- **Configuration Register**

  - Control Bits
    - **ONE-SHOT – Controls one-shot conversion**
    - **ADC RES – Selects 9 – 12-bit resolution**
    - **SHUTDOWN – Places device in low-power mode**
  - Not used in this class
    - **FAULT – Req'd. # of faults before asserting ALERT pin**
    - **POL – Controls ALERT pin polarity**
    - **INT – Selects Interrupt or Comparator mode**

Hands On!

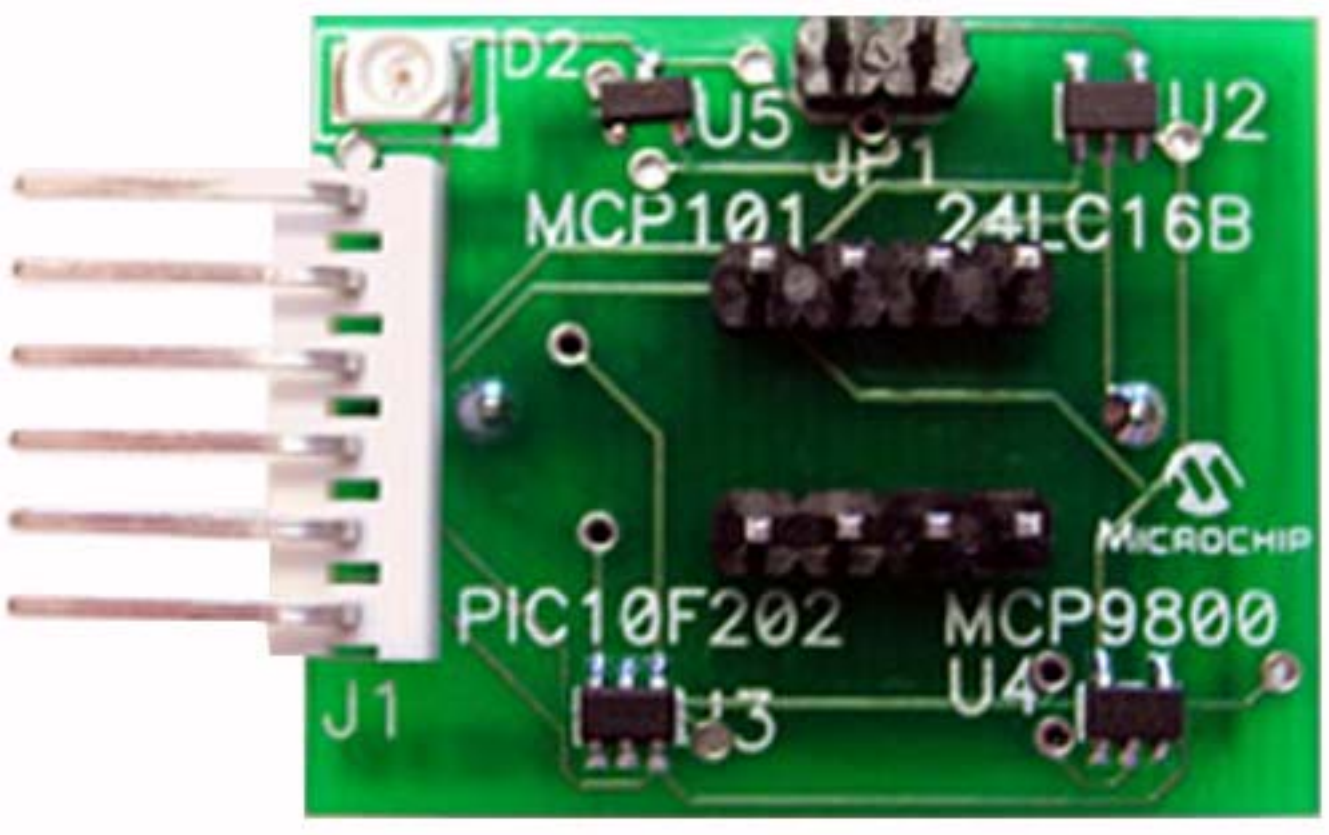
Documentation

- **You should have the following on your desk, which you can keep**
  - Lab Worksheet
  - Temperature logger demo board
  - EEPROM Data Book on CD
- **You will also have the following for reference documents in the class directory on the Hard Drive**
  - PIC10F2XX Datasheet
  - Temperature sensor Datasheet
- **You will also have the following hardware in front of you**
  - PICkit™ Starter Kit Programmer
  - SEEVAL® 32 Evaluation System Hardware

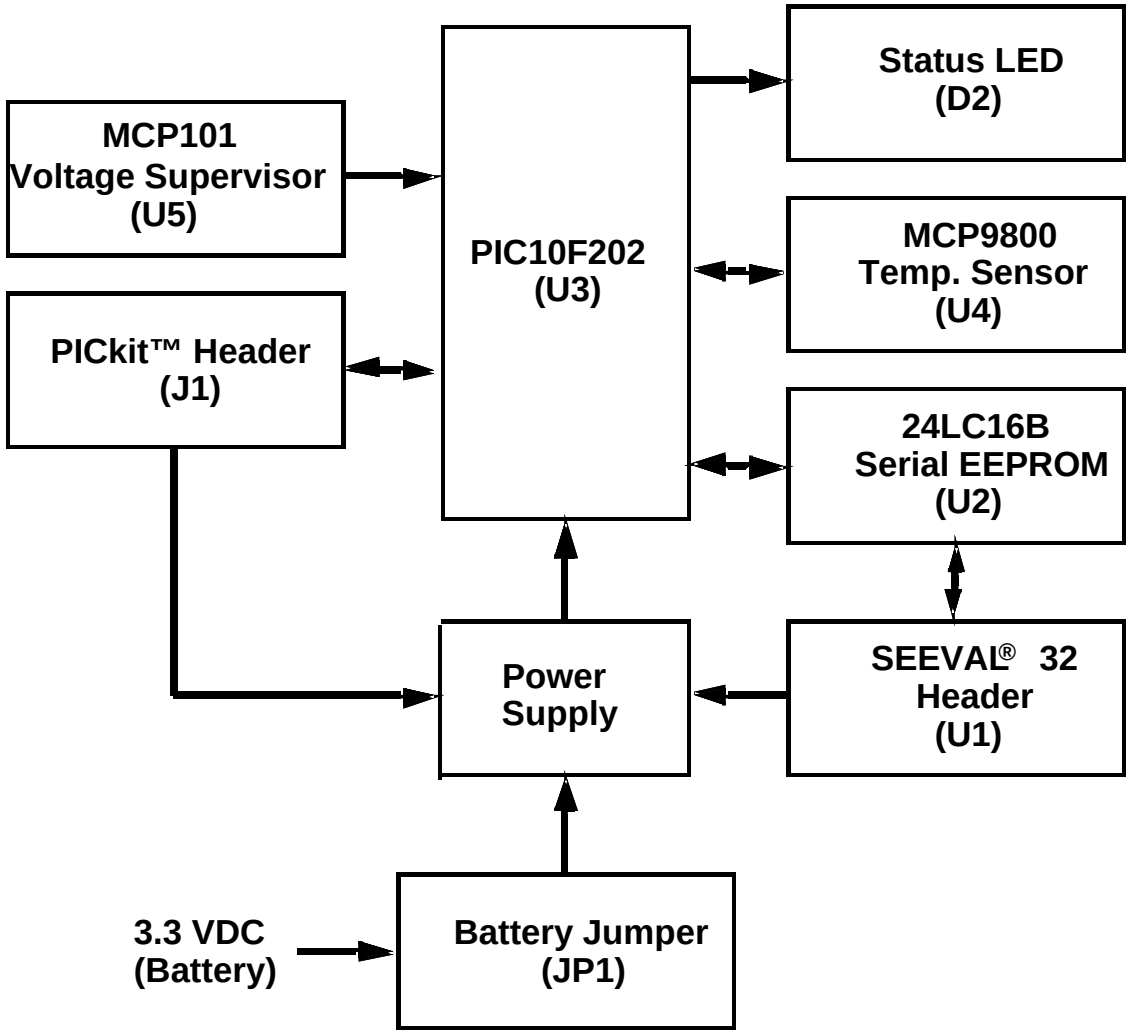
Hands-On Overview

- **Required to write code in Assembly for the PIC10F to complete a temperature logger**
  - Need to code a Serial EEPROM Write sequence
  - Need to code a sequence to read the temp sensor and write this to the EEPROM
- **Program the PIC10F using the PICkit™ Starter Kit**
- **Test the software by reading the E2 Memory**
  - SEEVAL® 32 Evaluation System Hardware and Software is provided
- **Lab Handout contains further instructions and schematics**

Temperature Logger Board



PC Board Overview

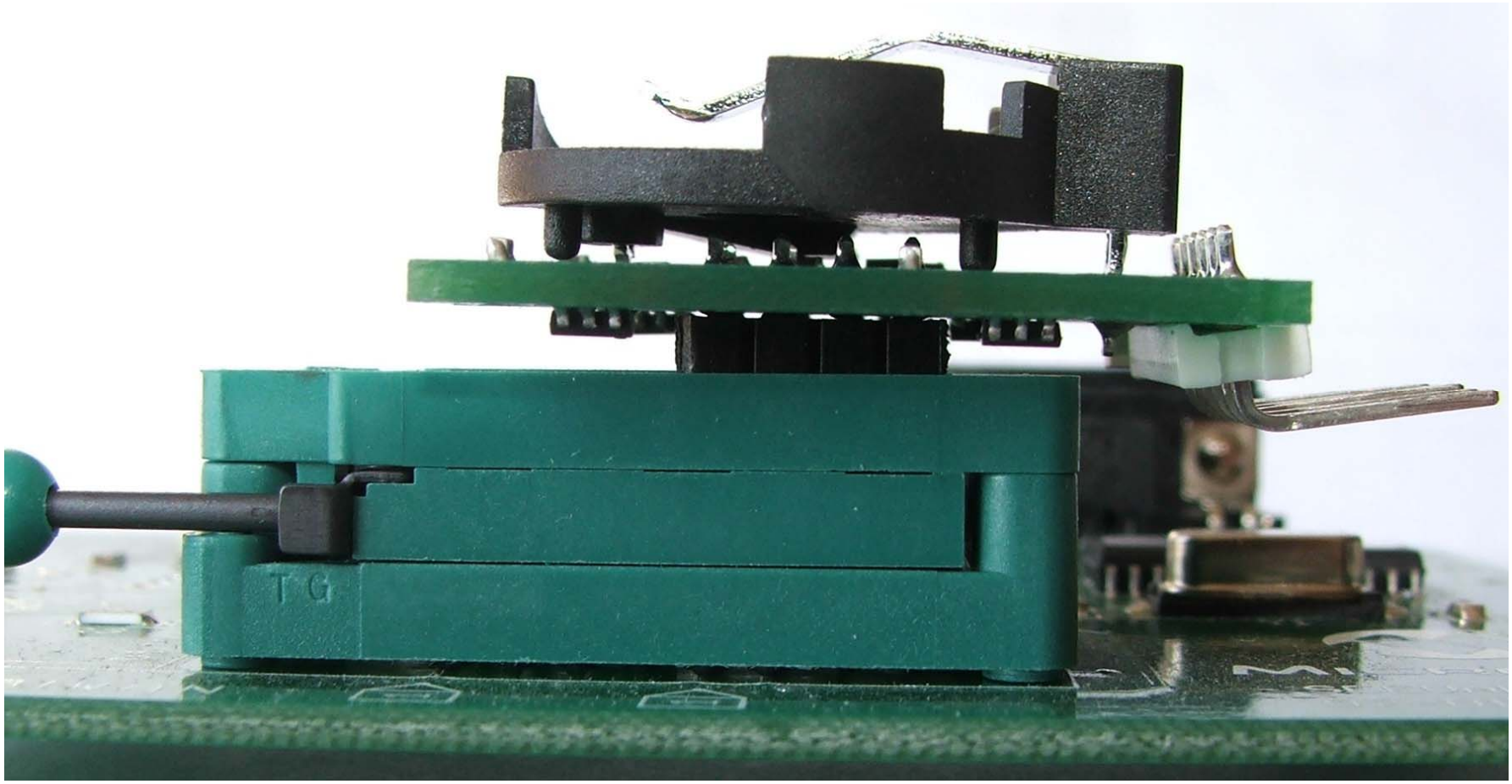


Attaching the PICkit™ Starter Kit to the Sensor Board

Using the PICkit™ Starter Kit to Program

- Start the PICkit programmer application by clicking on the icon on the desktop
- Select the PIC10F202 from the device list
- Import your generated HEX file
- Click on the write device button
- Disconnect your board from the PICkit Starter Kit
- Connect the battery and jumper if not already installed

Attaching the Sensor to the SEEVAL® 32 Evaluation System



Using SEEVAL® 32 Evaluation System

- Start SEEVAL 32 Evaluation System by clicking on the icon on the desktop
- Select I²C™ as the protocol
- Select 24AA16/… as the device
- Place the board in the socket as noted in the LAB handout
- Click on “Entire Device” under the read device block
- Check that the location and data that you have written are correct

Hardware Description

- Serial Data (SDA) is on GPIO.0
- Serial Clock (SCL) is on GPIO.1
- Status LED is on GPIO.2
- Jumper to disconnect onboard battery to reset firmware
- 8-Pin DIL header to interface with SEEVAL® 32 Evaluation System Hardware
- PICkit™ Starter Kit 6-Pin interface
- MCP101 is used to inhibit I²C™ writes when connected to the SEEVAL 32 Hardware

Provided Software Functions

- BSTART – Generate Start condition
- BSTOP – Generate Stop condition
- SEND_ACK – Generate Acknowledge
- SEND_NACK – Generate Not Acknowledge
- BYTEOUT – Output byte to bus
- BYTEIN – Input byte from bus

Code to Write to EEPROM

- **Exercise 1**
- **Write data to a location in Serial EEPROM**
  - Generate START
  - Send control byte including block select bits (3 MSb of EEPROM address) for Write
  - Send LSB of address
  - EEPROM Data Byte
  - Generate STOP
- **You can test your code by writing to an EEPROM location and reading using the SEEVAL® 32 Evaluation System Hardware**

Code to Read Temperature Sensor

- **Exercise 2**
- **Read converted data from the Temperature Register into a byte; the LSB is ignored to give us an 8-bit value**

  Generate START

  Send Control Byte for Write

  Send Temperature Register Address

  Generate START

  Send Control Byte for Read

  Read data from BUS

  Send NACK to end operation

  Generate STOP

Testing Your Code

- When you have completed Exercise 2
- Reset the board by removing and replacing the jumper
- After the LED begins blinking, warm the sensor with your finger
- Freeze spray is available
- Read the data back using the SEEVAL® 32 Evaluation System
- Export your data by going to File->Export so you can graph it using Excel

Graphing the Temperature

- Open the Excel spreadsheet by clicking on the icon on the desktop
- Choose to enable macros, if asked
- Click on the ‘Import Data’ button
- Select the file you just exported with SEEVAL® 32 Evaluation System and click OK
- Enter your sampling period (in seconds)
- Click on OK

Hands-On Summary

- Know the terminology of the I²C™ Bus
- Know the sequence of bus transactions to complete a Serial EEPROM write operation
- Know how to read from an external I²C device
- Know how to integrate I²C into an application
- Be familiar with PICkit ™ Starter Kit and SEEVAL® 32 Evaluation System Hardware

Further Modifications

- Change the time between temperature recordings
- Use the full 12-bit resolution of the temperature sensor
- Implement robust I²C™ error handling and fault detection
- Use data verification when writing to the EEPROM

UNIVERSITY OF MICROCHIP UOFM MASTERS 2007

References

- 24LC16B Datasheet – DS21703
- MCP9800 Datasheet – DS21909
- PIC10F202 Datasheet – DS41239
- "Interfacing I²C™ Serial EEPROMs to PIC10 and PIC12 Devices" – AN982
- "Recommended Usage of Microchip I²C™ Serial EEPROM Devices" – AN1028
- SEEVAL® 32 - DV243002

- Please fill in your class questionnaire before leaving

Thank you!

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.