EEPROM

- Electrically Erasable Programmable Read Only Memory
  - Non volatile memory (contents maintained when power removed)
  - Single byte reads and writes
  - Multibyte reads and writes
    - Sequential reads and writes on a single page
  - Limited life
    - Maximum number of times can be reprogrammed (written to)
    - Typically in the range of 1,000,000 programming cycles
    - Not used as a replacement to RAM, due to limited number of times can be reprogrammed
    - Microchip (2014) has 1Mbit SPI & I2C EEPROMs for less than $2.50
    - Microchip (2014) has 64Mbit SPI Serial Flash for less than $3 (lower endurance than EEPROM – 100,000 cycles)

EEPROM

- Limited life due to accumulation of electrons during writes
  - Reduces the voltage difference between 0 and 1 until the device can no longer determine the difference
  - Typical life is 1,000,000 writes
  - Doesn't seem to be a limit on reads
  - Data retention is typically guaranteed for 10 or more years
    - The Microchip EEPROM I mentioned on the last slide claims > 200 year retention
    - I don't expect to be around to validate that claim

EEPROM

- Interface
  - Serial
    - SPI and I2C are two common serial interfaces (are others)
    - 8 pins or less
    - SPI
      - Write enable (WREN)
      - Write disable (WRDI)
      - Read status register (RDSR)
      - Write status register (WRSR)
      - Data read (READ)
      - Write data (WRITE)
  - Parallel
    - Typically 8 bit wide bus
    - 28 pins or more
    - Can be faster than serial, but require high number of pins

Flash Memory

- A type of memory developed from EEPROM
- Erasure of data must be performed in blocks
  - Versus single bytes being erasable with EEPROM
- Reads and writes can be performed on single bytes
- On a freshly erased block, all bytes are writable and readable
- Once a byte has been written to, there are limitations to future writes (possible values that can be written into the byte) until the block it is in is erased
- Example
  - If block erasure makes all bits 1 in the block
  - Any bits that are changed to 0 via a write, cannot be changed back to 1 until the block is erased
- The software that manages I/O for the device (such as filesystem) deals with this restriction

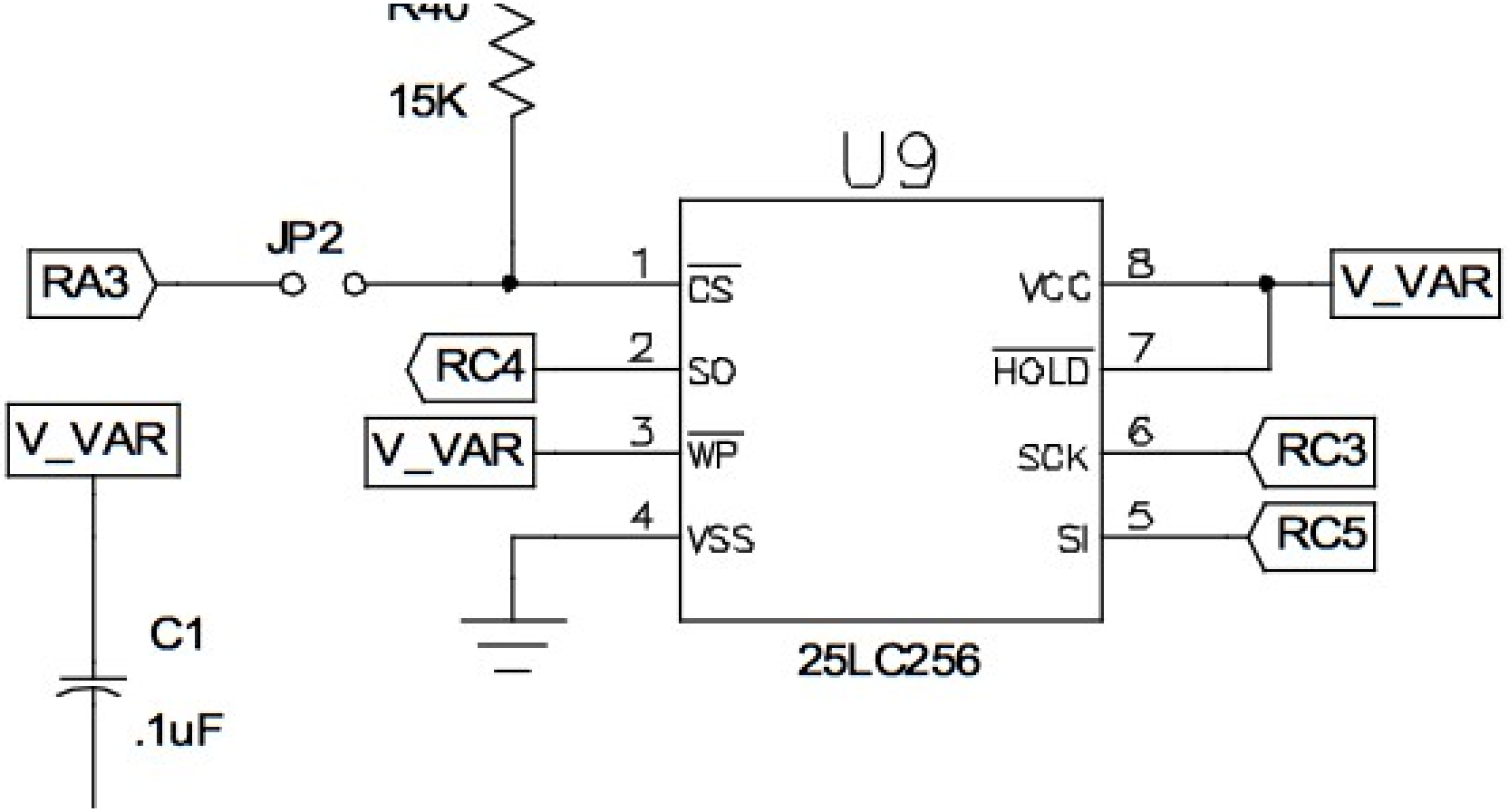
EEPROM on our Boards

- Both of our boards have serial EEPROM
  - PIC18 has SPI interface to EEPROM
    - 25LC256 (32K x 8), 256Kbit
    - Datasheet on blackboard
  - DEM 2 has I2C interface to EEPROM
    - 24LC256 (32K x 8), 236Kbit
    - Datasheet on blackboard

PIC18 25LC256

- SPI interface
- Maximum of 10 MHz clock
- 32,768 x 8 bit (256 Kbit)
- 64 byte pages
- 1,000,000 erase/write cycles
- 8 lines
  - RC3 – SCK, serial clock
  - RC4 – SDO, serial data output
  - RC5 – SDI, serial data input
  - RA3 – $\overline{CS}$, chip select

PIC18 25LC256



From PICDEM PIC18 Explorer Demonstration Board User's Guide

PIC 18 25LC256

- Read operation
  - Set $\overline{CS}$ low, selecting the device
  - Send 8 bit op code for read (READ)
  - Send MSB of 16 bit address
  - Send LSB of 16 bit address
    - Valid addresses 0x7FFF – 0x0000
  - Set $\overline{CS}$ high to terminate read
  - After each byte is transmitted from the EEPROM to the microcontroller, the address in the EEPROM is incremented to the next byte (rolls over from 0x7FFF to 0x0000)
    - Allows sequential reads

PIC18 25LC256

- Write operation
  - Set $\overline{CS}$ low, selecting the device
  - Send 8 bit op code to set the write latch (WREN)
  - Set $\overline{CS}$ high to end the command
  - Set $\overline{CS}$ low to start the write sequence
  - Send 8 bit op code for write (WRITE)
  - Send MSB of 16 bit address
  - Send LSB of 16 bit address
    - Valid addresses 0x7FFF – 0x0000
  - Send 8 bit value to be written at the address
  - Set $\overline{CS}$ high to terminate the write
  - Up to 64 bytes can be written in a single write command if they are on the same page

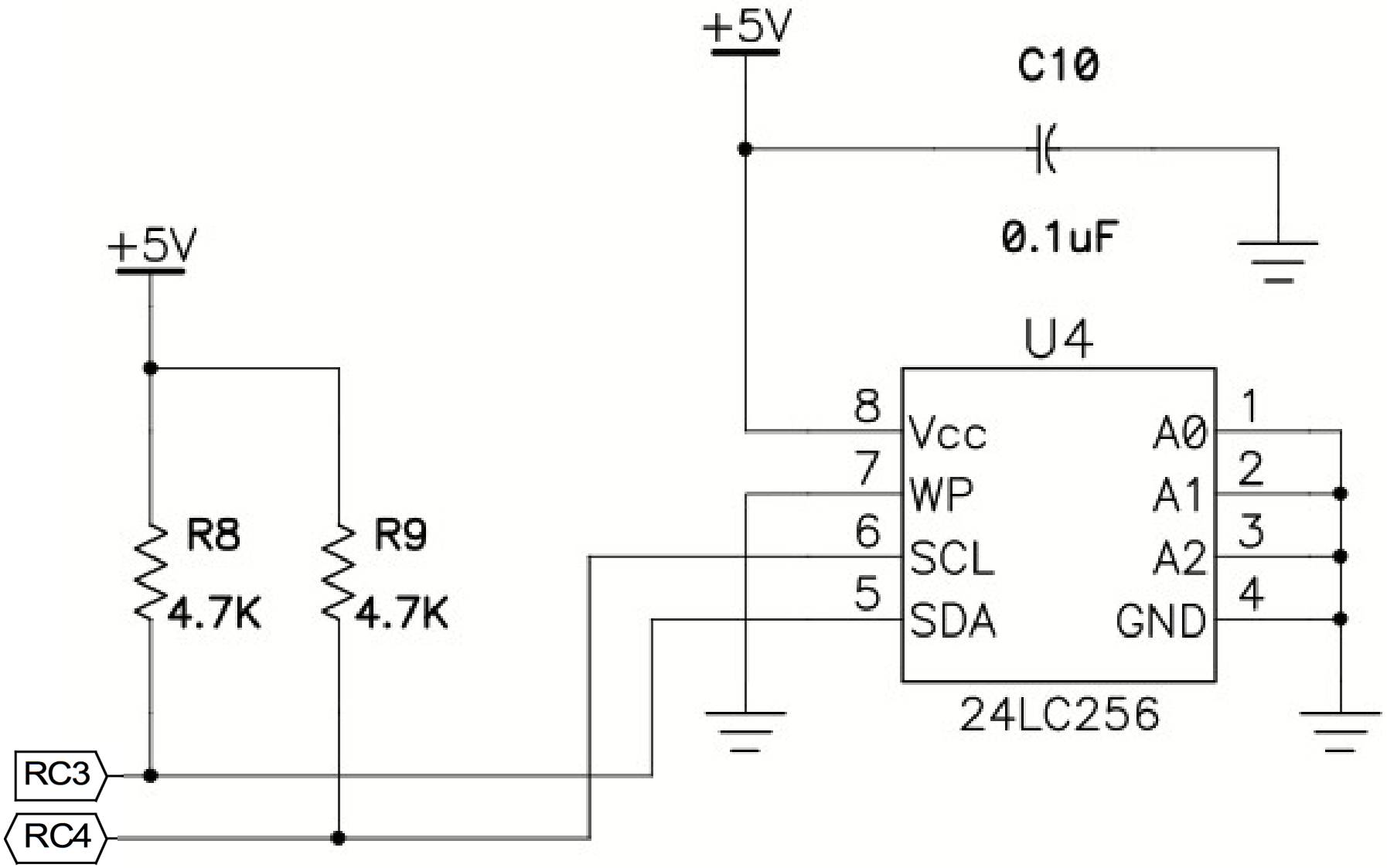
PIC18 25LC256

- Available commands
  - Read data from selected address
  - Write data to selected address
  - Reset the write enable latch (disable writing)
  - Set the write enable latch (enable writing)
  - Read status register
  - Write status register
- The EEPROM supports write protection setting via the write status register
- What does send XXX mean?
  - Clear SSPxIF flag
  - Copy 8 bits of data to send to the device to SSPxBUF
  - Poll SSPxIF until cleared

DEM 2 24LC256

- I2C interface
- Maximum of 1 MHz clock
- 32,768 x 8 bit (256 Kbit)
- 64 byte pages
- 1,000,000 erase/write cycles
- 8 lines
  - RC3 – SCL, serial clock
  - RC4 – SDA, serial data

DEM 2 24LC256



From PICDEM 2 Plus Demonstration Board User's Guide

I2C and EEPROM

- Initialization
  - Port direction and analog/digital
  - SSP1CON1, SSPSTAT, SSP1CON3
  - Set baud rate with SSPADD
- For reads, the EEPROM reads from the current address and increments the address pointer
- For writes, the EEPROM sets the address and then performs the write
- To read from the EEPROM
  - Set the address using beginning of write command
  - Once the address is set, restart the command sequence
  - Send the read command

I2C in General

- The general format for communicating via I2C is
  - Initialize the interface
  - Set the baud rate
  - Transfer data with the device
    - Send the start condition
    - Send the address of the device to communicate with
      - Last bit specified if reading or writing data
    - Receive acknowledgement from device
    - Transmit or receive data
      - Intermixed with acknowledgement
    - Send stop condition

Read From EEPROM (I2C)

- When polling SSPxIF
  - Include count limit
- To check acknowledgement
  - Check ACKSTAT bit of SSPCON2 register is not set
- If either the count limit is reached or the ACKSTAT bit is set
  - Write error message to the LCD
  - Initialize the I2C connection to the EEPROM
  - Pause
  - Return

I2C EEPROM Initialization

- Set SCL and SDA to digital input
- Set SSPCON1
  - Enable SSP
  - Enable I2C master mode, clock = Fosc/(4(SPADD+1))
- Set SSPSTAT
  - Disable slew rate control
- SSPCON3
  - Enable interrupt on detection start and stop conditions
- SSPADD
  - Example, 100kHz baud rate with 4MHz oscillator
    - 100kHz = $(4 \times 10^6)/(4(SPADD+1))$
    - SPADD = $(((4 \times 10^6)/4)/(100 \times 10^3)) - 1 = 9$

Read From EEPROM (I2C)

- Example
  - Send start condition (SEN)
    - Set SEN to 1
    - Poll SSPxIF
    - Check ASCSTAT bit is 0
  - Example code

```
while( (!PIR1bits.SSP1IF) && (count++ < limit) );
    if( (SSP1CON2bits.ACKSTAT) || (count >= limit) )
    {
        errorEEPROM(1);
        return 1;
    }
```

Write to EEPROM (I2C)

- Send start condition (SEN)
- Send device address with write enabled
- Send MSB of address to write
- Send LSB of address to write
- Send value to be written
- Send stop condition
- Pause 5ms to ensure write complete
  - Without the pause I got errors when writing large number of bytes in a loop