Kiler Ridge Olive Farm

The farm employs a college intern over the summer to work with the owner on one of his side projects for the farm

Gregg Bone: Caltech grad 78'; founder of multiple companies; owner of Kiler Ridge

← He was my mentor for the summer

My Summer Project

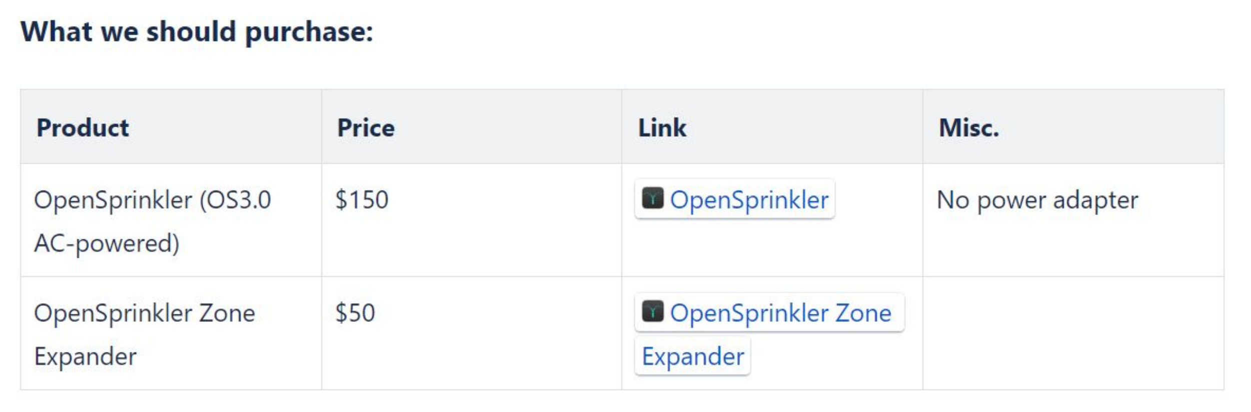
Goal: Design a system that can manage the irrigation on the farm and alert the owner via email if there there is a malfunctioning sprinkler.

Possible Irrigation Malfunctions:

- The water pump's output is **higher** than the threshold
  - Ex. A rat bit through the line
- The water pump's output is **lower** than the threshold
  - Ex. The switch for the pump is manually shut off

Approach

- To purchase the newest version of the open source sprinkler system



What we should purchase:

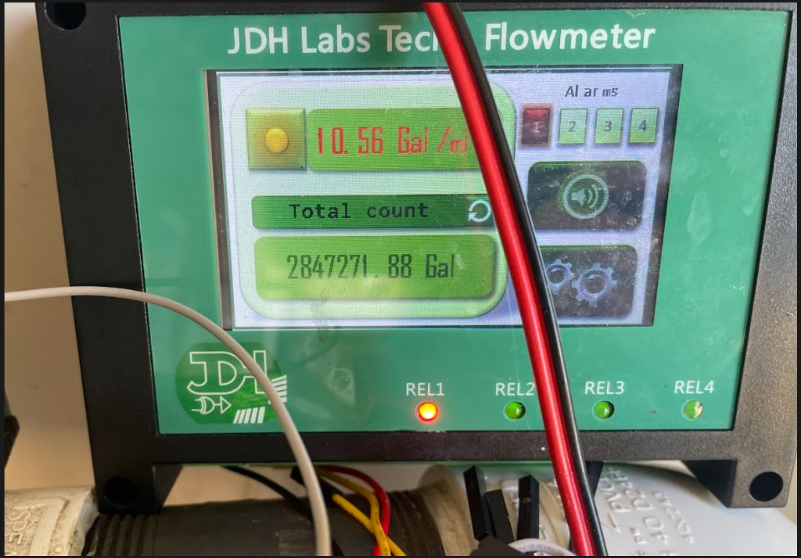
Product	Price	Link	Misc.
OpenSprinkler (OS3.0 AC-powered)	$150	OpenSprinkler	No power adapter
OpenSprinkler Zone Expander	$50	OpenSprinkler Zone Expander	

Important Features:

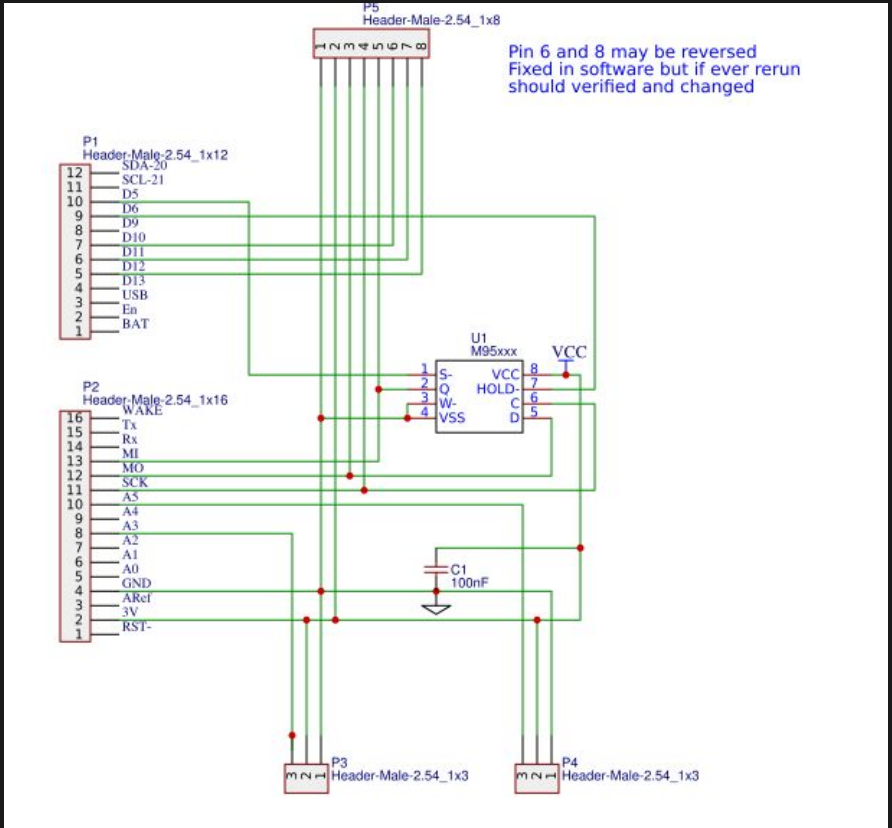
- current sensing - multi zones - wired ethernet - built in wifi - rain and flow sensors capability

Flow Sensor

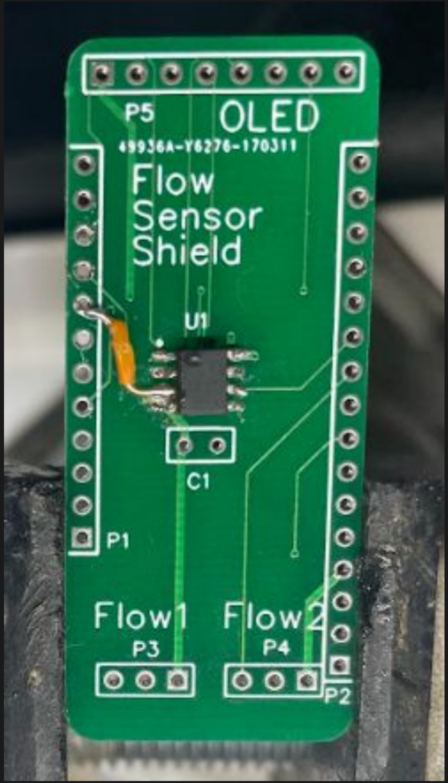
Goal: Develop an IOT device that would read the flow rate of the sensor and be able to communicate to the open sprinkler device if it is abnormal

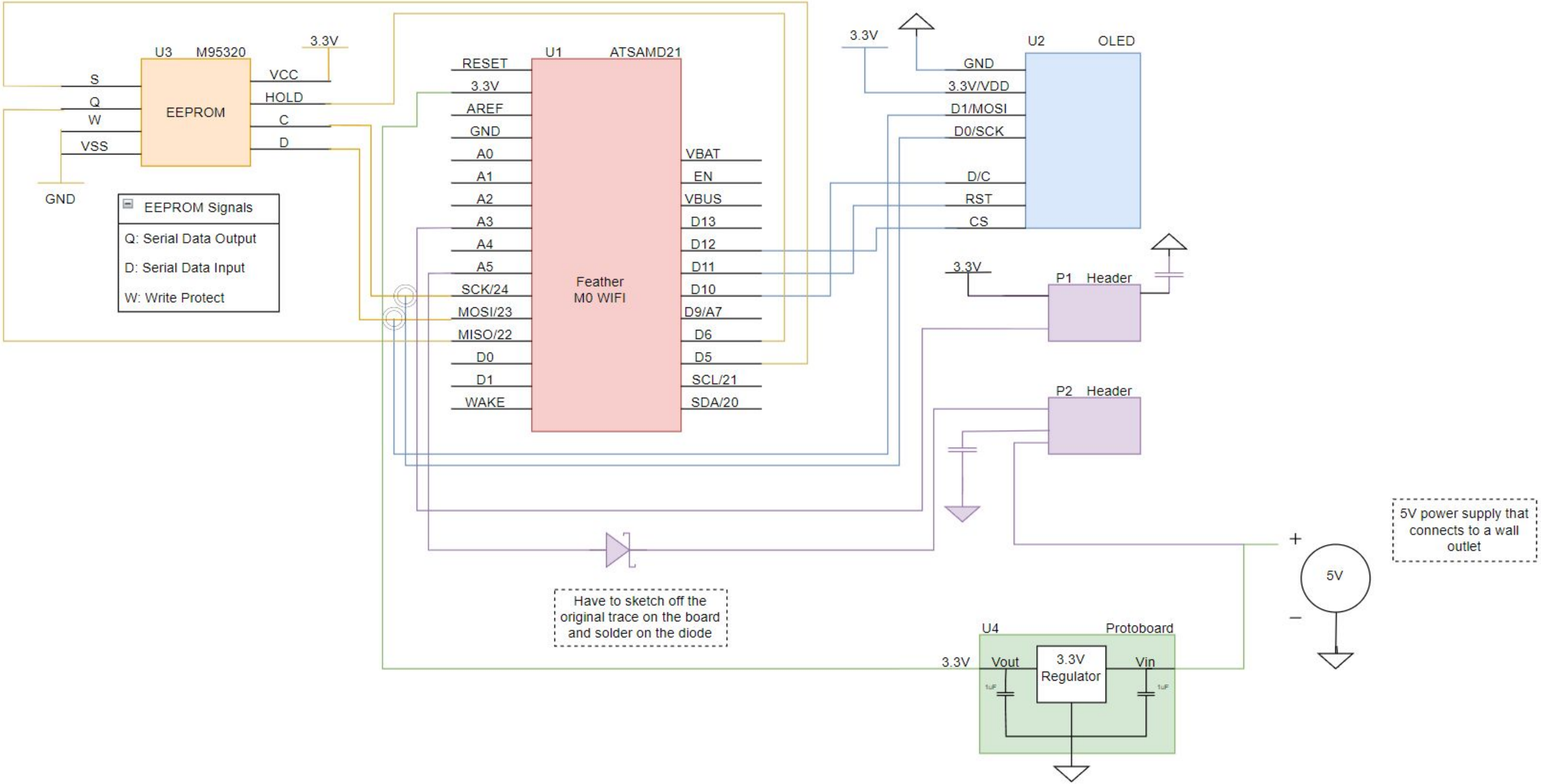


Previously Designed Flow Sensor



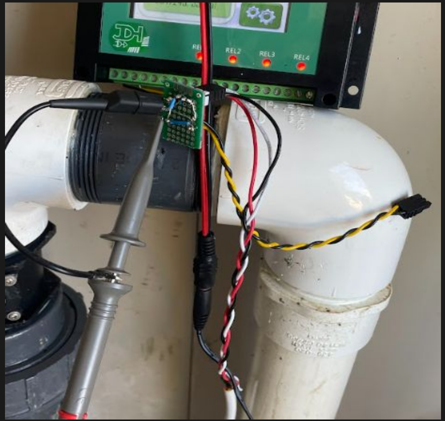
Pin 6 and 8 may be reversed
Fixed in software but if ever rerun
should verified and changed

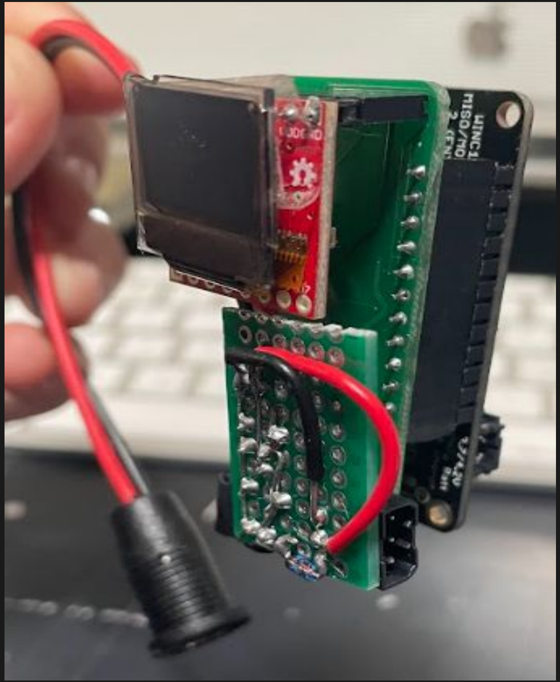


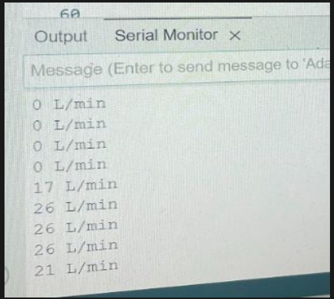


Newly Designed Flow Sensor

First Prototype







Message (Enter to send message to 'Ada

```
0 L/min
0 L/min
0 L/min
0 L/min
17 L/min
26 L/min
26 L/min
26 L/min
21 L/min
```

YF-DN50 Data Sheet

Horizontal test pulse frequency(Hz) = 0.2*Q +-3% (Q=L/min)

How to find the equation flow of water

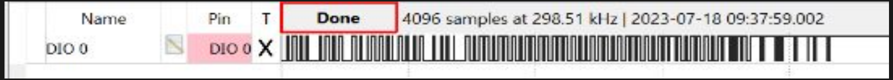
$1L/sec = (0.2)(L/min)$

$1L/sec = (0.2)(1L/min * 1min/60sec)$

$V_{total}(L) = N * 1/303L$

$N/t = 303*Q(L/s)$

$Q(L/min) = f*60/303 = f/5.05$

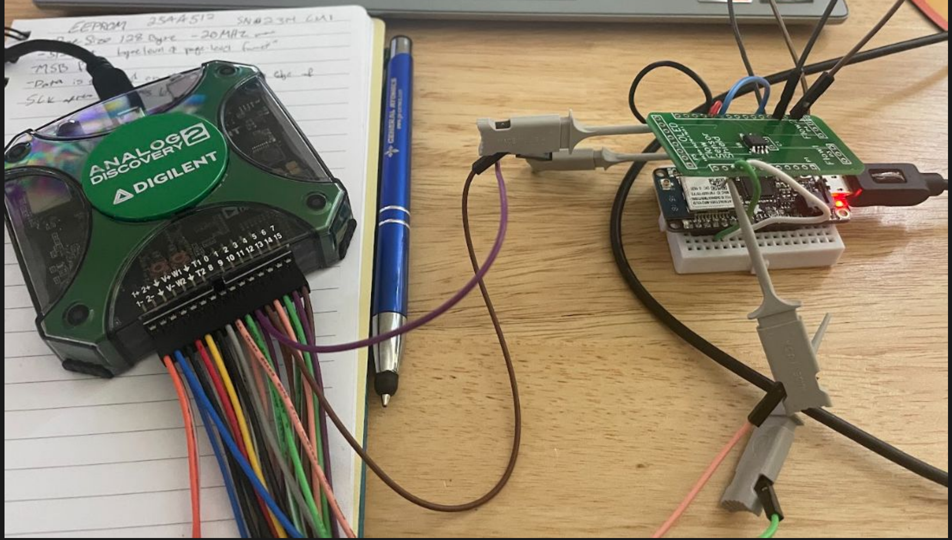


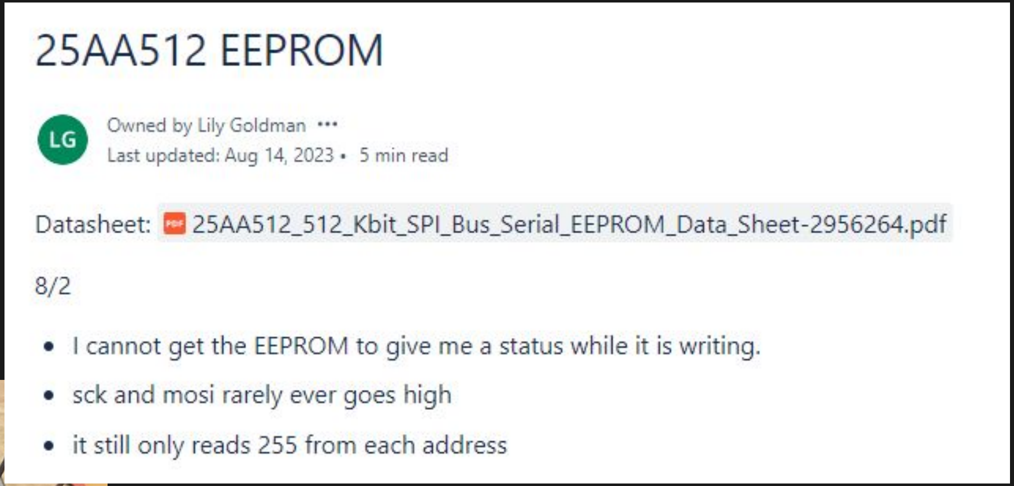
4096 samples at 298.51 kHz | 2023-07-18 09:37:59.002

DIO 0 | DIO 0 | X

Debugging the EEPROM

- An EEPROM is external storage that can be erased and reprogrammed.
- It is useful because it can be constantly written to and will never lose the data





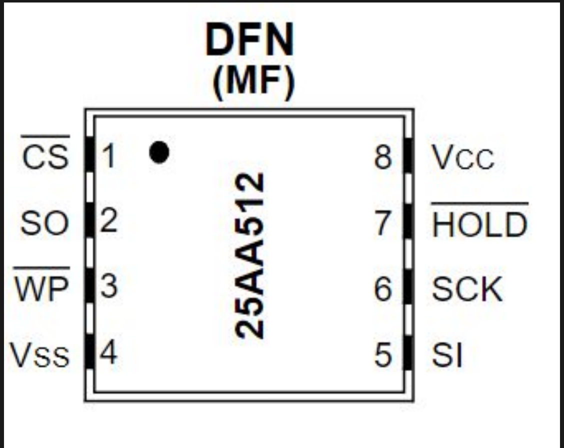
25AA512 EEPROM

Owned by Lily Goldman ···
Last updated: Aug 14, 2023 • 5 min read

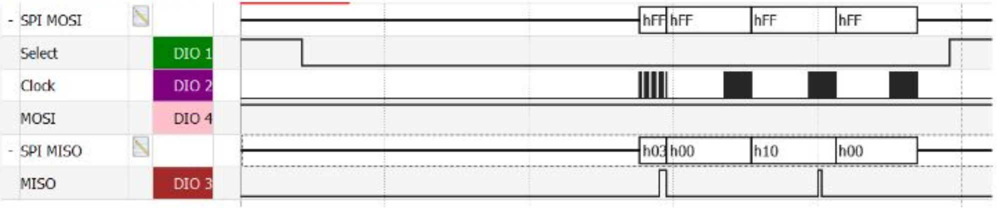
Datasheet: 25AA512_512_Kbit_SPI_Bus_Serial_EEPROM_Data_Sheet-2956264.pdf

8/2

- I cannot get the EEPROM to give me a status while it is writing.
- sck and mosi rarely ever goes high
- it still only reads 255 from each address

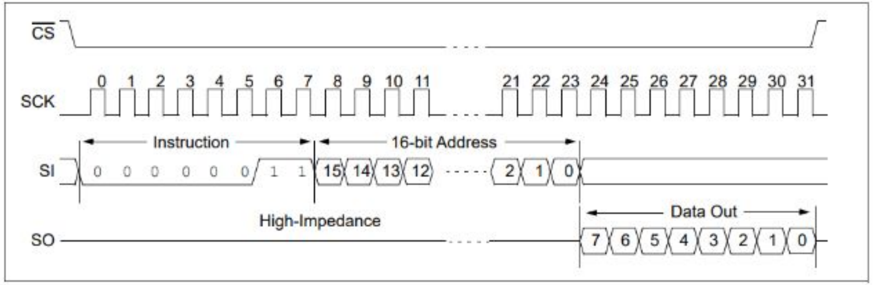


4. Read EEPROM: addr 16



a.

FIGURE 3-1: READ SEQUENCE



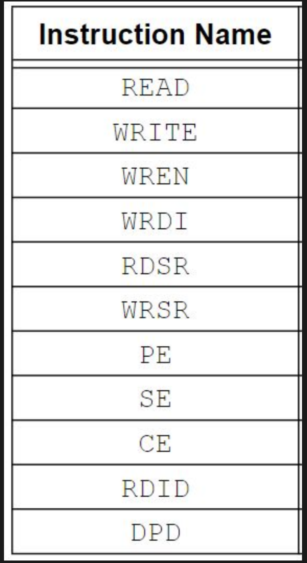
b.

c.

```
digitalWrite(EE_CS, LOW);
SPI.transfer(3);
SPI.transfer(addr>>8);                                    // first address byte
SPI.transfer(addr & 0xff);                                   // second address byte
byte data = SPI.transfer(0);

digitalWrite(EE_CS, HIGH);
Serial.print("Read data: ");
Serial.println(data);
```

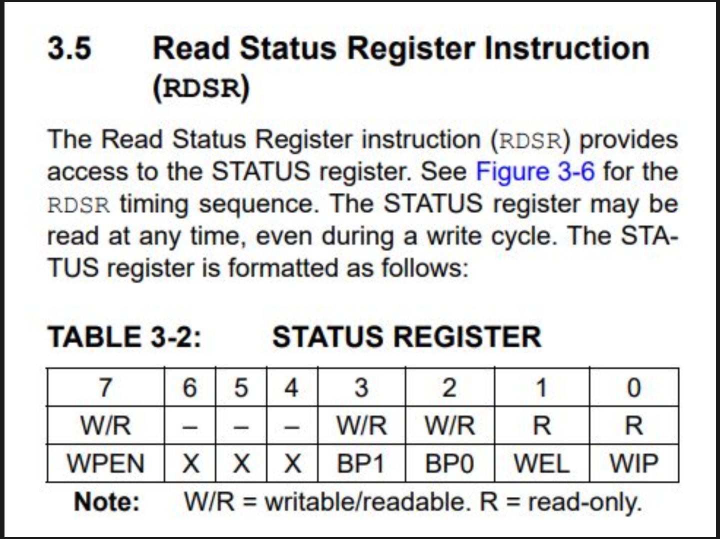
I repeated this exact documentation for each necessary instruction to verify they match the documentation.



Instruction Name
READ
WRITE
WREN
WRDI
RDSR
WRSR
PE
SE
CE
RDID
DPD

Every address would read 255 no matter what was written. So I needed to keep digging.

Debugging the EEPROM cont.



3.5 Read Status Register Instruction (RDSR)

The Read Status Register instruction (RDSR) provides access to the STATUS register. See Figure 3-6 for the RDSR timing sequence. The STATUS register may be read at any time, even during a write cycle. The STATUS register is formatted as follows:

TABLE 3-2: STATUS REGISTER

7	6	5	4	3	2	1	0
W/R	–	–	–	W/R	W/R	R	R
WPEN	X	X	X	BP1	BP0	WEL	WIP

Note: W/R = writable/readable. R = read-only.

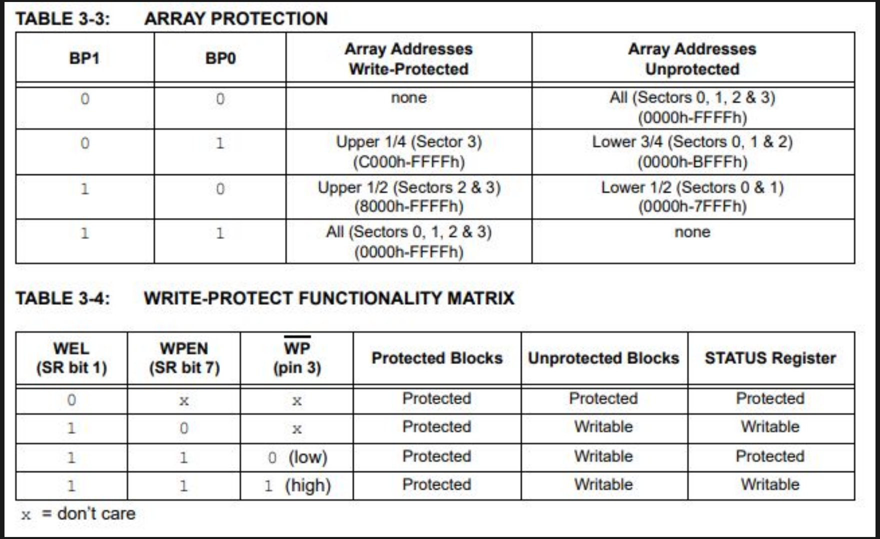


TABLE 3-3: ARRAY PROTECTION

BP1	BP0	Array Addresses Write-Protected	Array Addresses Unprotected
0	0	none	All (Sectors 0, 1, 2 & 3) (0000h-FFFFh)
0	1	Upper 1/4 (Sector 3) (C000h-FFFFh)	Lower 3/4 (Sectors 0, 1 & 2) (0000h-BFFFh)
1	0	Upper 1/2 (Sectors 2 & 3) (8000h-FFFFh)	Lower 1/2 (Sectors 0 & 1) (0000h-7FFFh)
1	1	All (Sectors 0, 1, 2 & 3) (0000h-FFFFh)	none

TABLE 3-4: WRITE-PROTECT FUNCTIONALITY MATRIX

WEL (SR bit 1)	WPEN (SR bit 7)	$\overline{\text{WP}}$ (pin 3)	Protected Blocks	Unprotected Blocks	STATUS Register
0	x	x	Protected	Protected	Protected
1	0	x	Protected	Writable	Writable
1	1	0 (low)	Protected	Writable	Protected
1	1	1 (high)	Protected	Writable	Writable

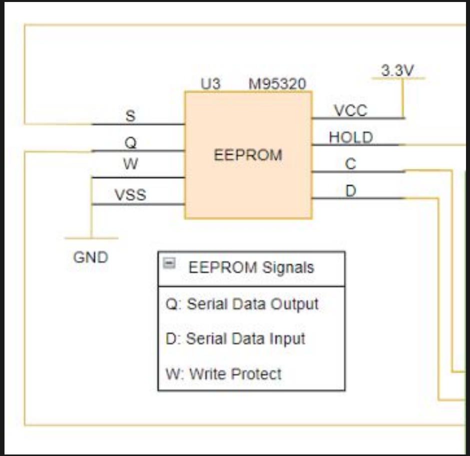
x = don't care

This is when I finally realised this was not a software issue, but a hardware issue. I couldn't write to the status register to make all addresses protected.

EEPROM Solution

The write protect was always tied low so all the addresses were protected and unable to be written to.

I had to take a small wire and tied it to a separate GPIO pin so I could set it high to access the addresses while writing.

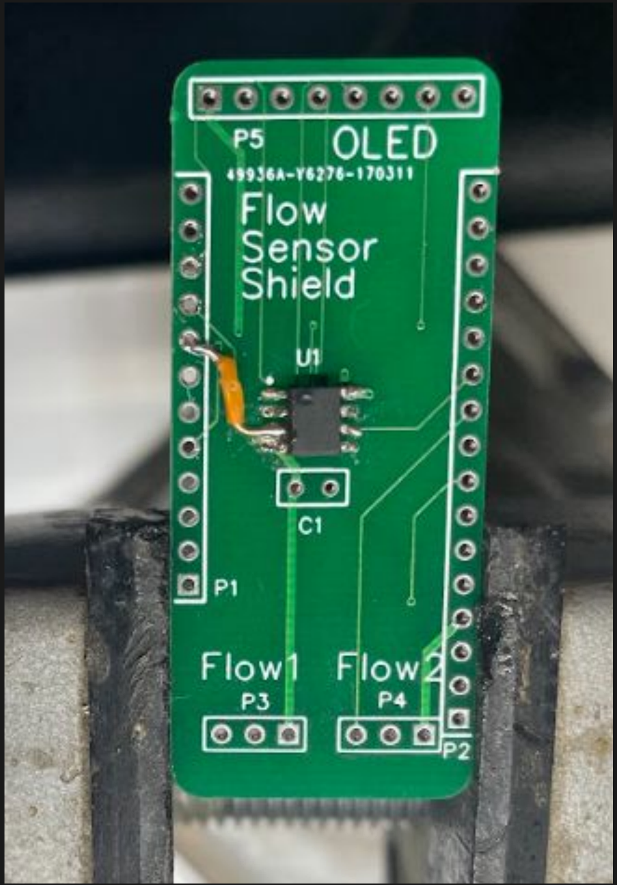


EEPROM Signals

Q: Serial Data Output

D: Serial Data Input

W: Write Protect



Presentation Recap

- The EEPROM worked!!!
- The flow sensing worked!!
- The product had over the air capabilities and could transmit to the OpenSprinkler device.
- The complicated part was updating the firmware to allow the OpenSprinkler to receive the data.