Whole Class Project Architecture

Last updated 8/2/21

Whole Class Project - Architecture

- What we know
  - Red LED signal
  - Pattern of 1's and 0's
- First thoughts
  - Use our photo-sensor to capture the LED signal
    - Will not be 1/0 at the output of the photo-sensor → A/D conversion
  - Set a threshold on the A/D output to determine if we have 1/0
    - This is often called a data slicer
  - Collect 1's and 0's until we have a letter (or a word)
  - Output the letter to the 7-segment display
- Early questions
  - Does our photo-sensor 'see' red LED light?
  - What threshold will we use for the data slicer?
  - How does the seven segment display work

Whole Class Project - Architecture

- Red LED
  - 620 – 680 nm wavelength

Electrical / Optical Characteristics at TA=25°C

Symbol	Parameter	Device	Typ.	Max.	Units	Test Conditions
λpeak	Peak Wavelength	Super Bright Red	660		nm	IF=20mA
λD [1]	Dominant Wavelength	Super Bright Red	640		nm	IF=20mA
Δλ1/2	Spectral Line Half-width	Super Bright Red	20		nm	IF=20mA
C	Capacitance	Super Bright Red	45		pF	VF=0V;f=1MHz
VF [2]	Forward Voltage	Super Bright Red	1.85	2.5	V	IF=20mA
IR	Reverse Current	Super Bright Red		10	uA	VR = 5V

- Photo-sensor
  - 550nm vs desired 650nm

Type	SEN-5001	
Light Resistance	@ 2FC	6 – 29
(kohms)	@1FC	12 - 58
R05 min. (kohms)	500	
Spectral Response	550nm	
Pmax (mW)	250	
Vmax (V)	250	

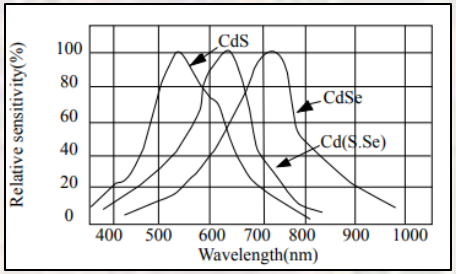


Chart for another sensor
No chart in our spec

!!!! NEED TO VERIFY THIS WILL WORK

Whole Class Project - Architecture

- LED – Photosensor Experiment
  - Setup a 1/0 pattern on a red LED
  - Shine it at out photo-sensor
  - Measure the photo-sensor with an A/D to make sure we have enough difference between a 1 and a 0

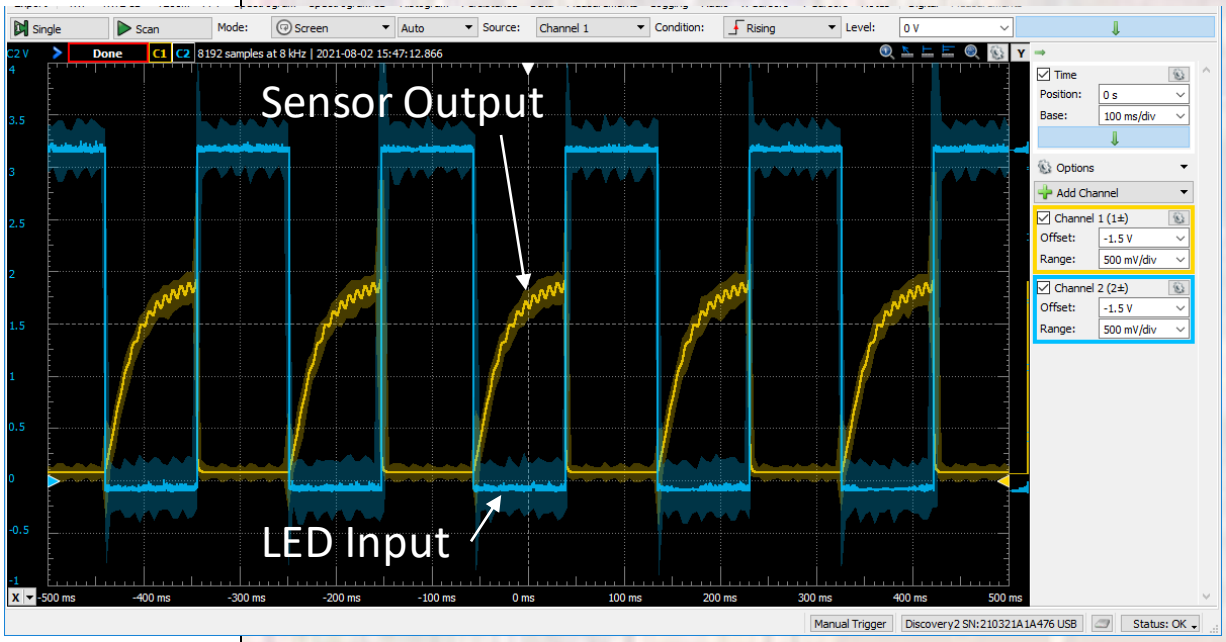
```
////////////////////////////////////
//
// project_whole_class_BMP project
//
// created 7/26/21 by tj
// rev 0
//
////////////////////////////////////
//
// Whole class project
//
// detector characterization
//
// flash an led and measure response
//
////////////////////////////////////

#include "mbed.h"
#include <stdio.h>

// Global HARDWARE Objects
// Create an ADC object, attached to A3
AnalogIn Photodet(A3);
// Create a digital output to drive the led
DigitalOut Led_out(D3);

int main_det(void){
    setbuf(stdout, NULL);

    // splash
    printf("Whole Class Project - detector_test - example for EE2905\n");
    printf("Using Mbed OS version %d.%d.%d\n\n",
           MBED_MAJOR_VERSION, MBED_MINOR_VERSION, MBED_PATCH_VERSION);
```

```
    // local variables
    float adc_val;
    int i;

    // initialize the LED to off
    Led_out = 0;
    adc_val = 0;

    // continuously check the detector and provide output
    while(1){
        // take 5 measurements per 'bit'
        for(i = 0; i < 10; i++){
            adc_val = Photodet.read();
            printf("%i - %f\n", Led_out.read(), adc_val);
            wait_us(25000/10)       // 40Hz -> 25ms loop with 10 samples / loop
        } // end for

        // toggle LED
        Led_out = !Led_out;

    }// end while

    return 0;
}// end main
```

Whole Class Project - Architecture

- LED – Photosensor Experiment
  - Not a very strong signal from the sensor

```
Whole Class Project - detector - example for EE2905
Using Mbed OS version 6.10.0

0 - 0.652259
0 - 0.663980
0 - 0.655433
0 - 0.635897
0 - 0.647619
1 - 0.023932
1 - 0.023443
1 - 0.023932
1 - 0.023199
1 - 0.014164
0 - 0.346032
0 - 0.458852
0 - 0.513065
0 - 0.534799
0 - 0.566300
1 - 0.023443
1 - 0.023443
1 - 0.024420
1 - 0.023443
1 - 0.023443
0 - 0.348718
0 - 0.460806
0 - 0.523321
0 - 0.543834
0 - 0.573138
1 - 0.023687
```

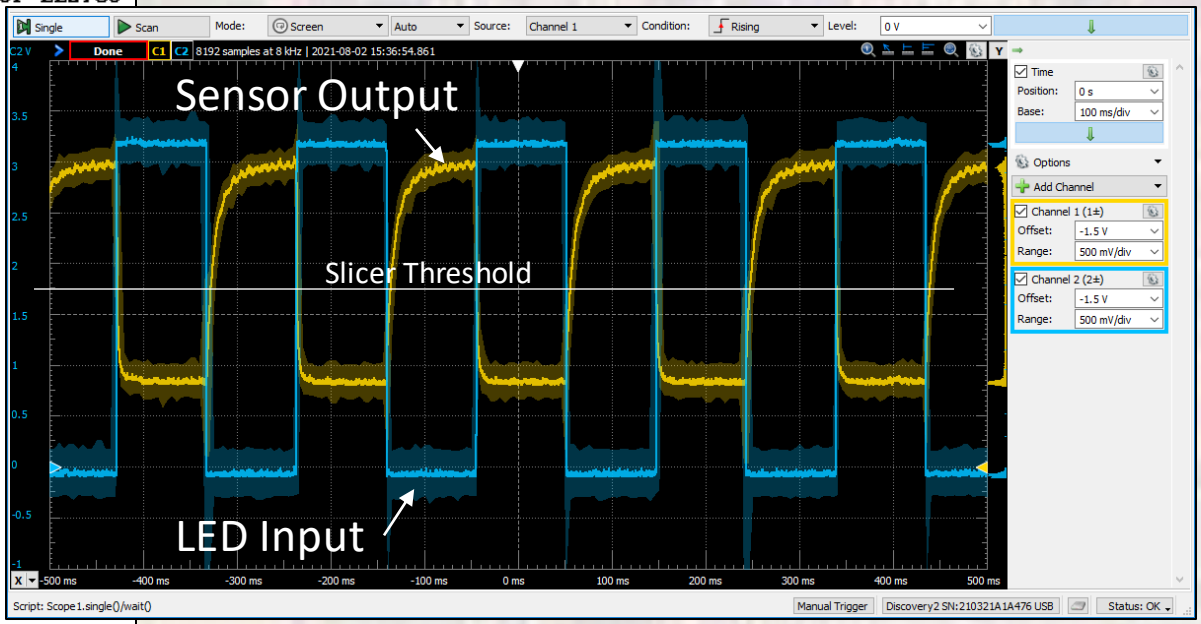


- Need a better answer

Whole Class Project - Architecture

- LED – Photosensor Experiment
  - Switch from 10KΩ load resistor to a 1KΩ load resistor
  - 0.22 for ‘on’
  - 0.82 for ‘off’

```
Whole Class Project - detector - example for EE2905
Using Mbed OS version 6.10.0

0 - 0.936996
0 - 0.931624
0 - 0.928694
0 - 0.933578
0 - 0.938462
1 - 0.225397
1 - 0.224420
1 - 0.223443
1 - 0.223443
1 - 0.222955
0 - 0.839072
0 - 0.876679
0 - 0.894506
0 - 0.910134
0 - 0.918926
1 - 0.224420
1 - 0.223687
1 - 0.223199
1 - 0.223199
1 - 0.222466
0 - 0.831014
0 - 0.871062
0 - 0.886203
0 - 0.902320
0 - 0.913309
1 - 0.224420
1 - 0.223443
1 - 0.223199
1 - 0.222955
1 - 0.222466
0 - 0.831990
0 - 0.870086
```



- Set the data slicer threshold to 0.52

Need to keep an eye on this signal swing and sharpness

Whole Class Project - Architecture

- Resource Management
  - 1 analog input
  - 8 digital outputs
  - This is easy
    - A3, and D 8,9,10,11,12,13,14,15
  - If we had other components, we would need to avoid conflicts – e.g. MISO and a digital output, PWM pin and digital input, …

Whole Class Project - Architecture

- LED – Photosensor Experiment
  - We now have a potential red LED interface circuit

    AND
  - a possible threshold value (0.52)

  - Looking at the waveform – we may want to 'oversample' the sensor to ensure better readings
    - Arbitrarily (for now) oversample at 5x
    - Make this modifiable in our solution

Whole Class Project - Architecture

- Data Structure
  - After some analysis, the data structure has a pattern
  - The information is in the 1's, the 0's are delimiters
  - 1 → "dot"
  - 111 → "dash"
  - Space between dots and dashes (0) → "break"

Individual letters are separated
By 3 0's in a row → "letter"

Individual words are separated
By 7 0's in a row → "word"

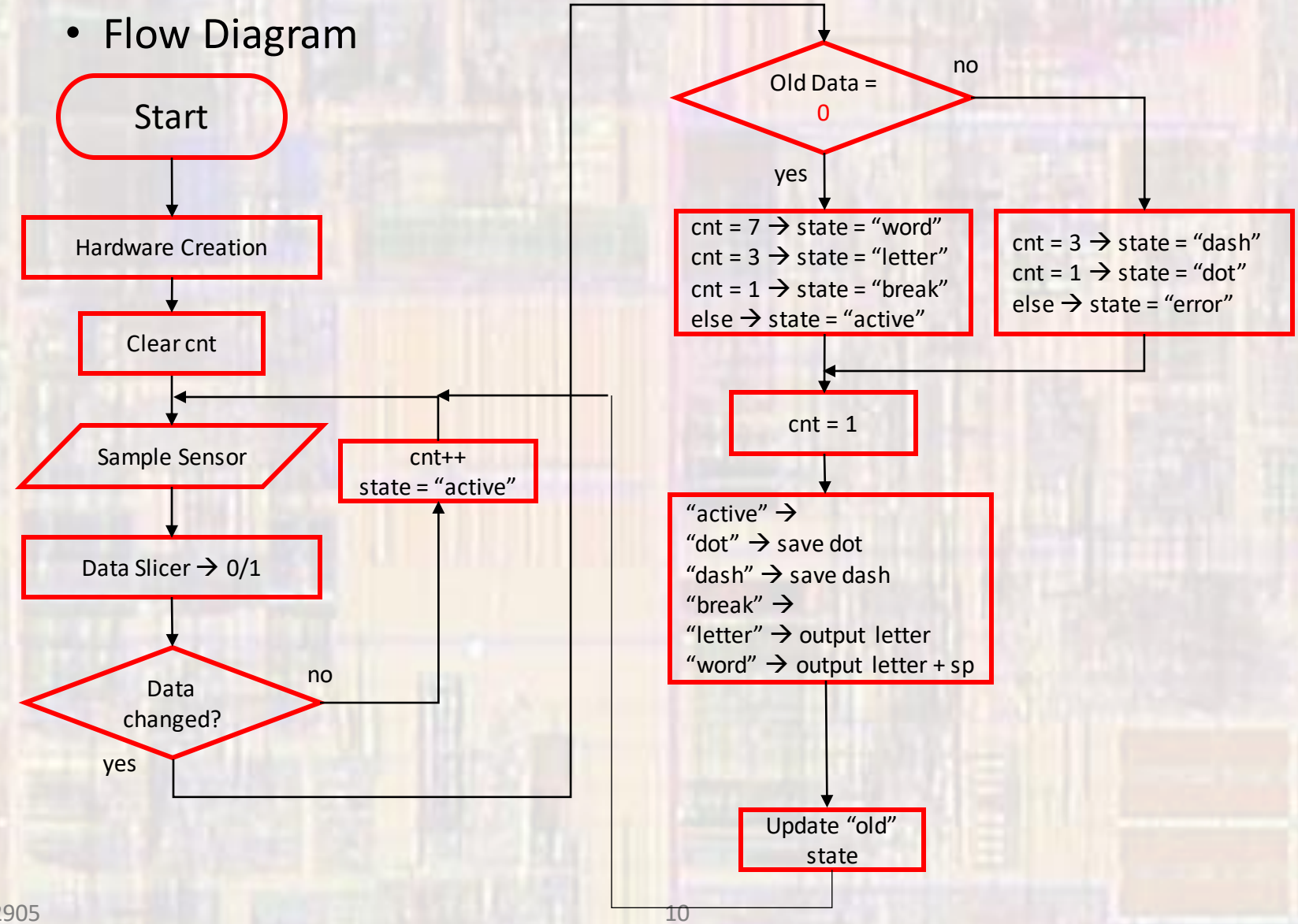
Baud rate = 40Hz
Each 1 or 0 lasts for 25ms

A	10111	•—	J	1011101110111	•———	S	10101	•••
B	111010101	—•••	K	111010111	—•—	T	111	—
C	11101011101	—•—•	L	101110101	•—••	U	1010111	••—
D	1110101	—••	M	1110111	——	V	101010111	•••—
E	1	•	N	11101	—•	W	101110111	•——
F	101011101	••—•	O	11101110111	———	X	11101010111	—••—
G	111011101	——•	P	10111011101	•——•	Y	1110101110111	—•——
H	1010101	••••	Q	1110111010111	——•—	Z	11101110101	——••
I	101	••	R	1011101	•—•			

- Amazing – we think in Morse Code !!!

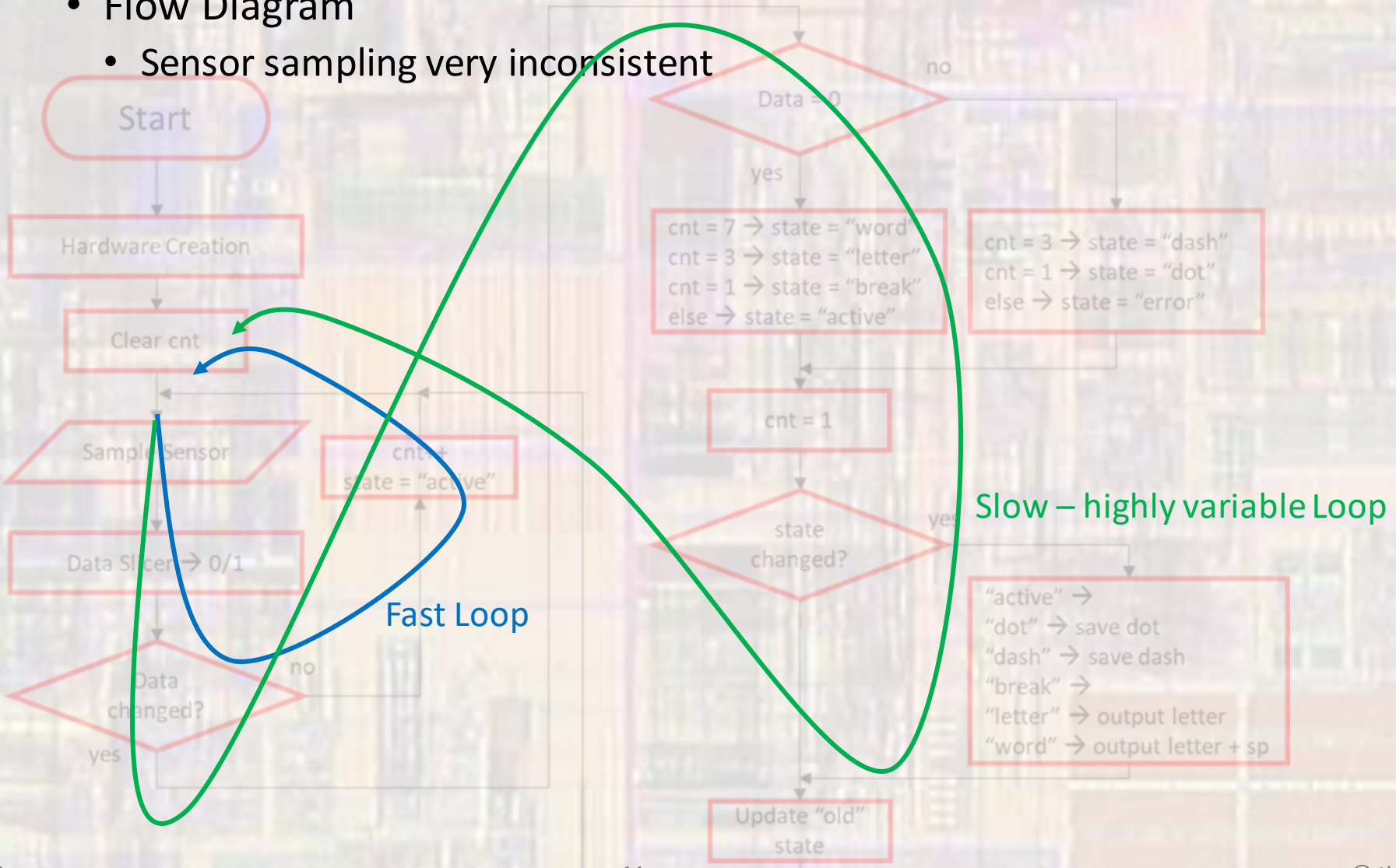
Whole Class Project - Architecture

- Flow Diagram



Whole Class Project - Architecture

- Flow Diagram
  - Sensor sampling very inconsistent



Whole Class Project - Architecture

- Improved Program Flow
  - Ensure sensor is sampled at a regular interval
    - Setup a Ticker
    - Use the Ticker ISR to sample sensor
    - Store the result into a global variable
  - Use the main loop to check the global variable for changes and take appropriate actions
    - As long as the main loop circulates faster than the sensor sample time we should get consistent results

Whole Class Project - Architecture

- Improved Flow Diagram
  - Use a Ticker to sample the sensor