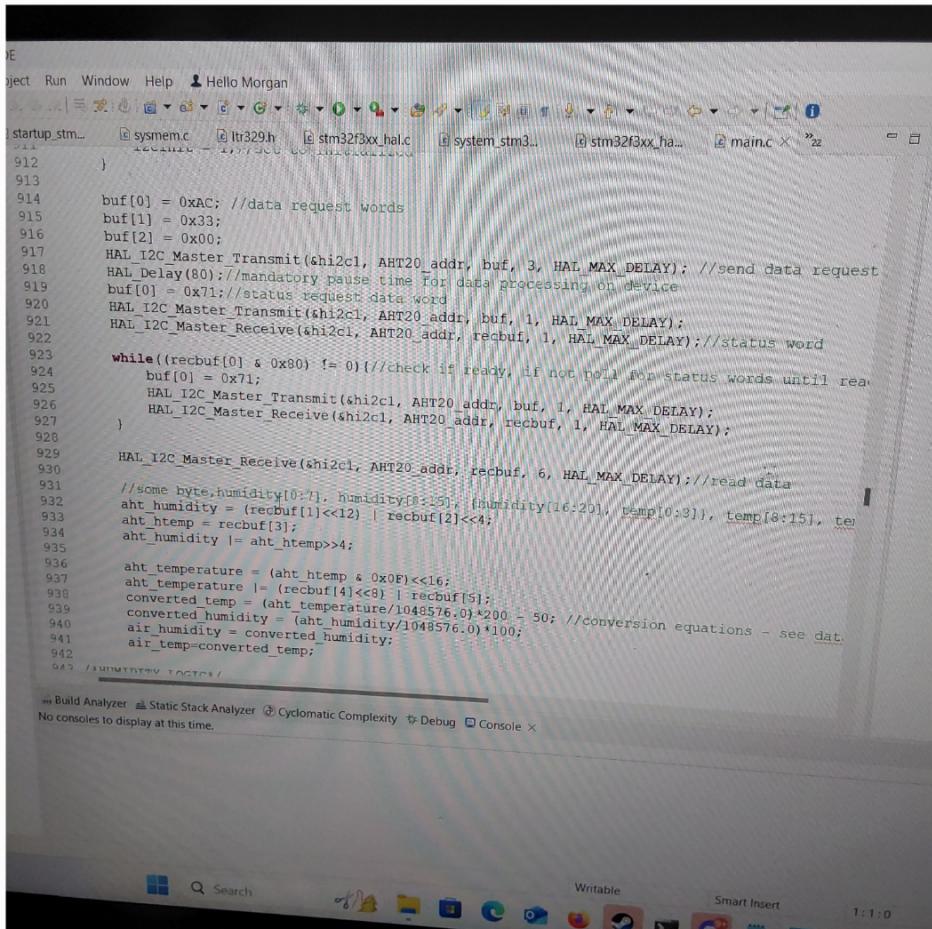


This weekend we did a lot of individual sensor testing primarily in communicating with the i2c sensors and float wpswitch

Here is some code, we also tested the i2c on a single bus



```
912     }
913
914     buf[0] = 0xAC; //data request words
915     buf[1] = 0x33;
916     buf[2] = 0x00;
917     HAL_I2C_Master_Transmit(&i2c1, AHT20_addr, buf, 3, HAL_MAX_DELAY); //send data request
918     HAL_Delay(80); //mandatory pause time for data processing on device
919     buf[0] = 0x71; //status request data word
920     HAL_I2C_Master_Transmit(&i2c1, AHT20_addr, buf, 1, HAL_MAX_DELAY);
921     HAL_I2C_Master_Receive(&i2c1, AHT20_addr, recbuf, 1, HAL_MAX_DELAY); //status word
922
923     while((recbuf[0] & 0x80) != 0){ //check if ready, if not poll for status words until ready
924         buf[0] = 0x71;
925         HAL_I2C_Master_Transmit(&i2c1, AHT20_addr, buf, 1, HAL_MAX_DELAY);
926         HAL_I2C_Master_Receive(&i2c1, AHT20_addr, recbuf, 1, HAL_MAX_DELAY);
927     }
928
929     HAL_I2C_Master_Receive(&i2c1, AHT20_addr, recbuf, 6, HAL_MAX_DELAY); //read data
930
931     //some byte, humidity[0:7], humidity[8:15], (humidity[16:20], temp[0:3]), temp[8:15], temp[16:20]
932     aht_humidity = (recbuf[1]<<12) | recbuf[2]<<4;
933     aht_htemp = recbuf[3];
934     aht_humidity |= aht_htemp>>4;
935
936     aht_temperature = (aht_htemp & 0x0F)<<16;
937     aht_temperature |= (recbuf[4]<<8) | recbuf[5];
938     converted_temp = (aht_temperature/1048576.0)*200 - 50; //conversion equations - see data sheet
939     converted_humidity = (aht_humidity/1048576.0)*100;
940     air_humidity = converted_humidity;
941     air_temp=converted_temp;
942
943 //HUMIDITY TO TEMP
944
945 //Build Analyzer  Static Stack Analyzer  Cyclomatic Complexity  Debug  Console
946 No consoles to display at this time.
```

Some basic TDS code was written too, we use an ADC for this and set it to polling. Originally an interrupt based approach would block out the other sensor code. So we opted for a adc poll which fixed the problems



```
counter++;
printf("collect");

if(counter<30){
    for(int i = 0; i<counter; i++){
        sum += (float)tdsbuff[i];
        printf("%d\n",tdsbuff[i]);
    }
    avg = sum/((float)counter);
}
else{
    for(int i = 0; i<30; i++){
        sum += (float)tdshuff[i];
        printf("%d\n",tdshuff[i]);
    }
    avg = sum/30.0;
}

float waterTemp = 25.0;
float c1 = 1.0 + 0.02*(waterTemp-25);
c1 = (avg*3.3/4096)/c1;
ppm = ((133.42*c1*cv*cv - 255.86*cv*cv + 857.39*cv)*0.5;
sum = 0;
TDS_sample_counter--;
if(TDS_sample_counter==0)
{
    initial_TDS=ppm;
}
/*FLOAT SWITCH SENSORS LOGIC - handled in interrupt handler*/
//Float switch pin -> high-too little water, low-good amount of water
```

