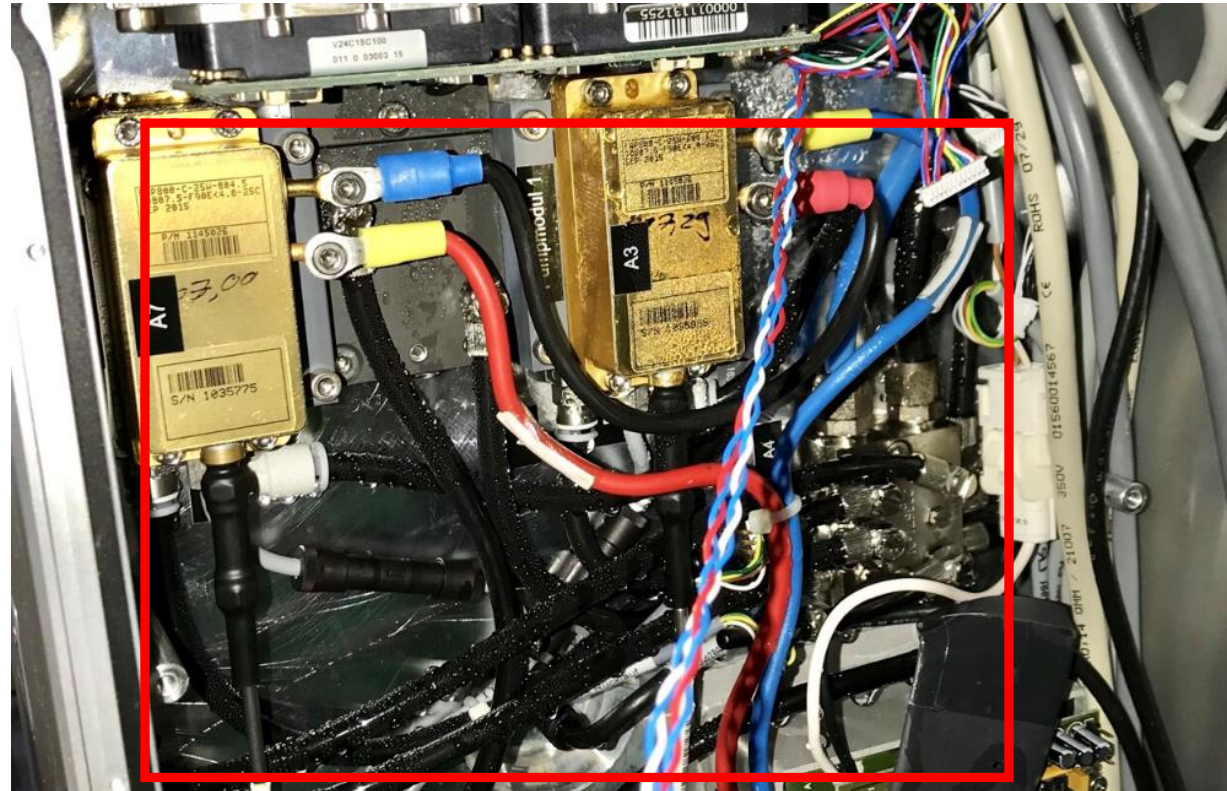
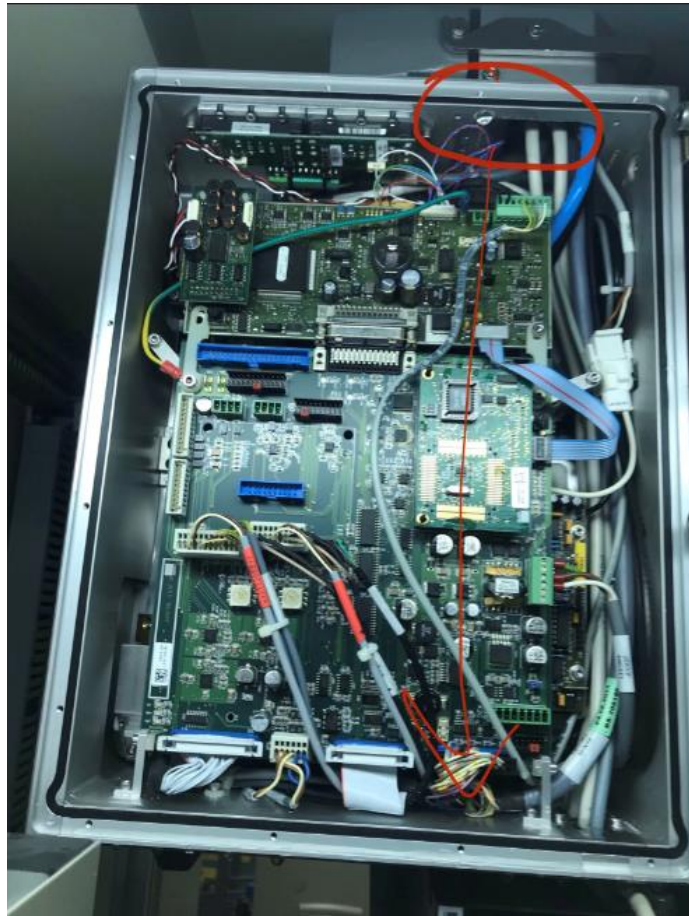


## Define the problem

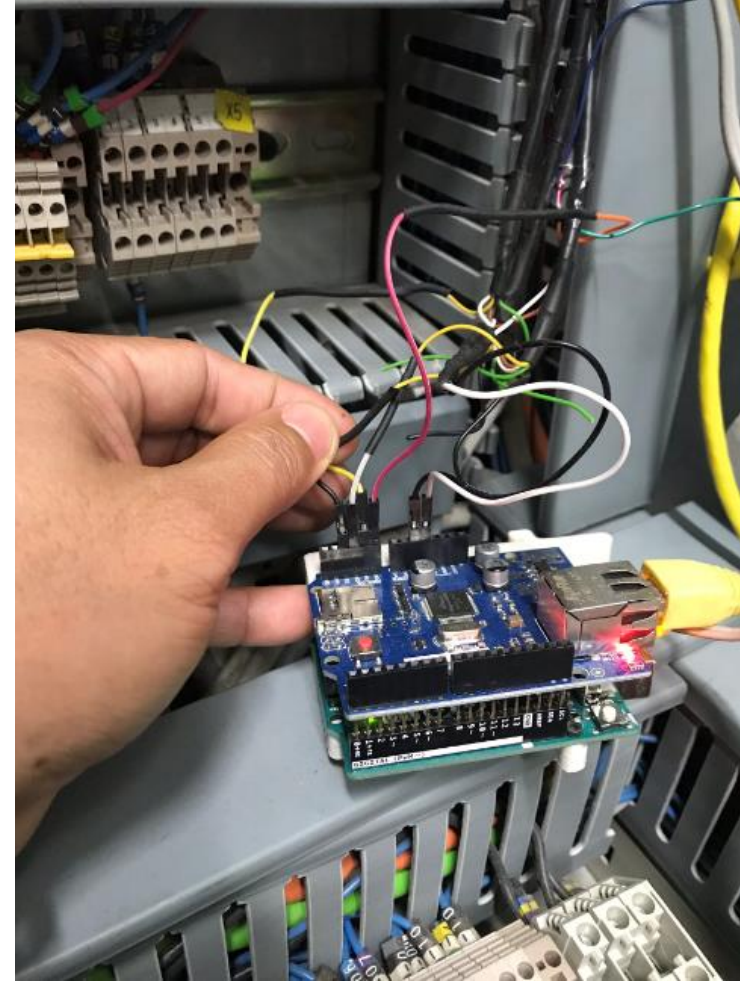
- Laser marking break down
- Take long time to repair > 7 day



## Root Cause Analysis

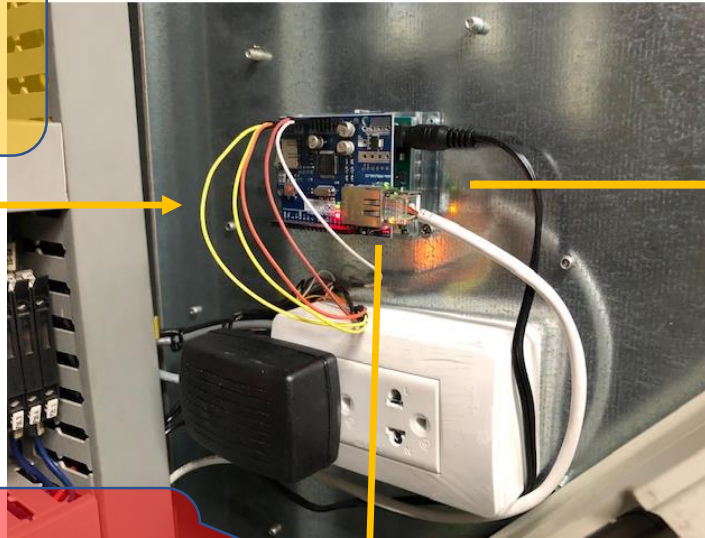
- Water leak from connector and water tube





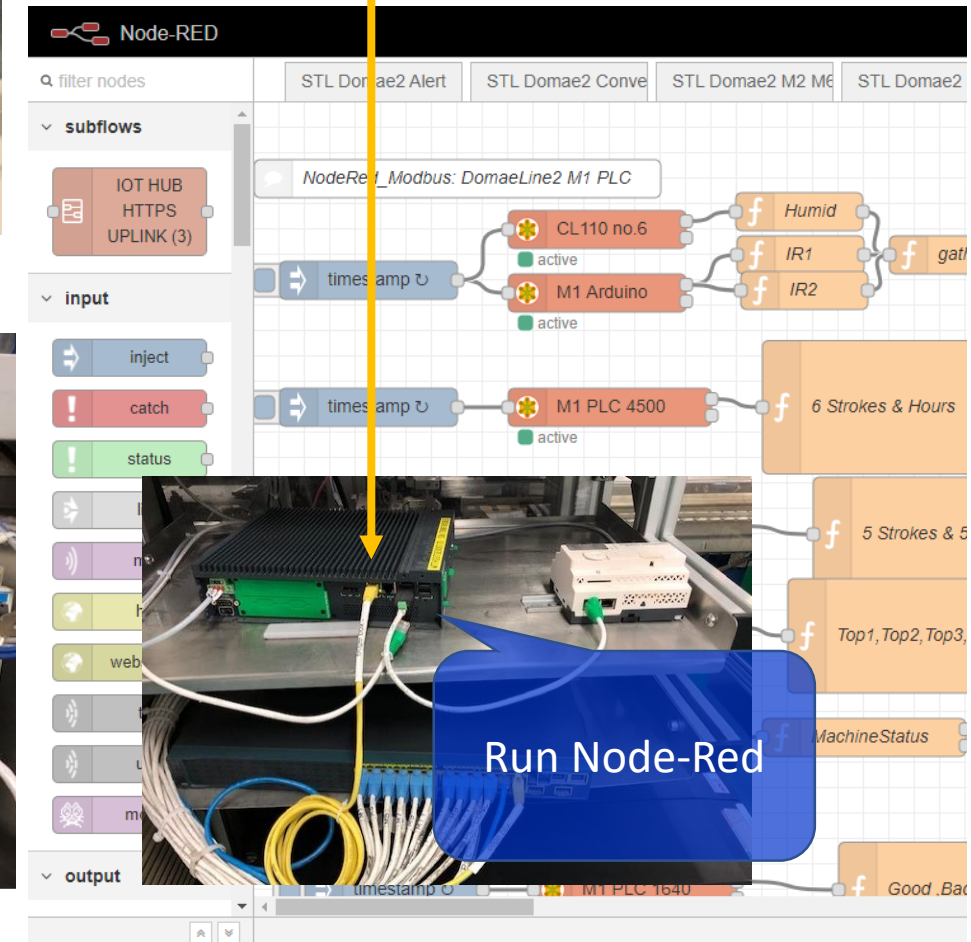
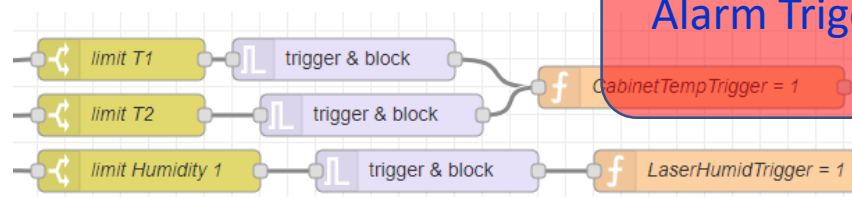


Read %RH from  
Sensor to  
Arduino Board



%RH save  
data on cloud

Alarm Trigger to PLC





## HTM2500LF

Temperature and Relative Humidity Module

### SPECIFICATIONS

- Hermetic Housing
- Humidity calibrated within  $\pm 2\%$  @55%RH
- Temperature measurement through NTC 10kOhms  $\pm 1\%$  direct output
- Small size product
- Typical 1 to 4 Volt DC output for 0 to 100%RH at 5Vdc

Based on the rugged HTS2035SMD humidity / temperature sensor, HTM2500LF is a dedicated humidity and temperature transducer designed for OEM applications where a reliable and accurate measurement is needed. Direct interface with a micro-controller is made possible with the module's humidity linear voltage output.



✓RoHS

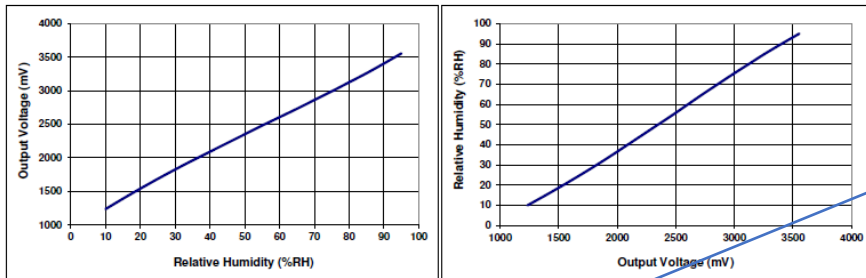
### TYPICAL PERFORMANCE CURVES

#### HUMIDITY SENSOR

Typical response look-up table ( $V_s = 5V$ )

RH (%)	Vout (mV)	RH (%)	Vout (mV)
10	1235	55	2480
15	1390	60	2605
20	1540	65	2730
25	1685	70	2860
30	1825	75	2990
35	1960	80	3125
40	2090	85	3260
45	2220	90	3405
50	2350	95	3555

Modeled linear voltage output ( $V_s = 5V$ )



#### Linear Equations

$$V_{out} = 26.65 \cdot RH + 1006$$

$$RH = 0.0375 \cdot V_{out} - 37.7$$

with  $V_{out}$  in mV and RH in %

#### Polynomial Equations

$$V_{out} = 1.05E^{-3} \cdot RH^3 - 1.76E^{-1} \cdot RH^2 + 35.2 \cdot RH + 898.6$$

$$RH = -1.92E^{-9} \cdot V_{out}^3 + 1.44E^{-5} \cdot V_{out}^2 + 3.4E^{-3} \cdot V_{out} - 12.4$$

with  $V_{out}$  in mV and RH in %

Domae3\_M1\_Humidity | Arduino 1.8.12

File Edit Sketch Tools Help

Domae3\_M1\_Humidity\$

```
#include <SPI.h>
```

```
#include <Ethernet.h>
```

```
#include "Mudbus.h"
```

Mudbus Mb; <- Define Modbus TCP

```
//Function codes 1(read coils), 3(read registers), 5(write coil), 6(write register)
```

```
//signed int Mb.R[0 to 125] and bool Mb.C[0 to 128] MB_N_R MB_N_C
```

```
//Port 502 (defined in Mudbus.h) MB_PORT
```

```
float Vout;
```

```
float RH;
```

```
void setup()
```

```
{
```

```
    delay(10000);
```

```
    uint8_t mac[] = { 0x90, 0xA2, 0xDA, 0x91, 0x49, 0x39 };
```

```
    uint8_t ip[] = { 192, 168, 2, 79 };
```

```
    uint8_t gateway[] = { 192, 168, 2, 1 };
```

```
    uint8_t subnet[] = { 255, 255, 255, 0 };
```

```
    Ethernet.begin(mac, ip, gateway, subnet);
```

```
    //Avoid pins 4,10,11,12,13 when using ethernet shield
```

```
    Serial.begin(9600);
```

```
}
```

```
void loop()
```

```
{
```

```
    Mb.Run();
```

```
    //convert to mV
```

```
    Vout = analogRead(A0);
```

```
    Vout = Vout*5000/1023;
```

```
    RH = 0.0375*Vout-37.7;
```

```
    Serial.print("Vout = ");
```

```
    Serial.print(Vout);
```

```
    Serial.print(", RH = ");
```

```
    Serial.println(RH,2);
```

```
    //Analog inputs 0-1023
```

```
    Mb.R[0] = int(RH); //pin A0 to Mb.R[0]
```

```
}
```

<- RH conversion

<- Map to Address 40001

Error downloading [http://arduino.esp8266.com/stable/package\\_esp8266com\\_index.json](http://arduino.esp8266.com/stable/package_esp8266com_index.json)

Disabled, Default 4MB with spiffs (1.2MB APP/1.5MB SPIFFS), 240MHz (WiFi/BT), QIO, 80MHz, 4MB (32Mb), 921600, None on COM5

# Domae Line 2 M01 Laser marking

3-Feb-2020

```
In [16]: import numpy as np
import pandas as pd
from matplotlib import pyplot

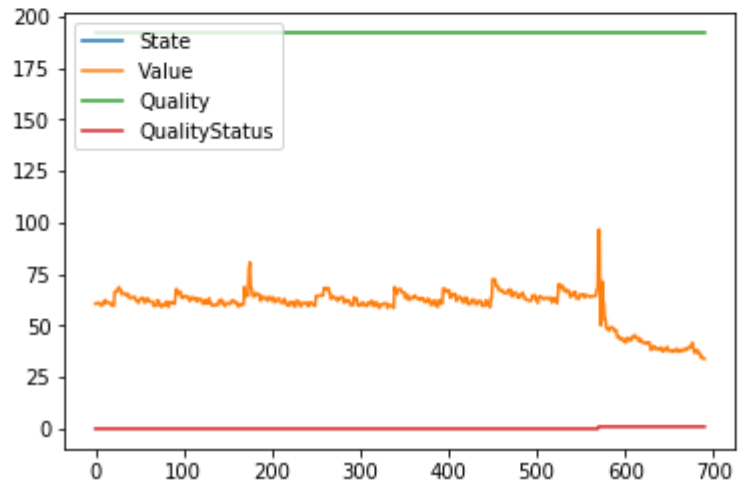
#data = read_csv('Insight Data 20200128_132638.csv')
#print(data.head())
#data.plot()
#pyplot.show()

data = pd.read_csv('Insight Data 20200128_132638_data_train.csv')
#data.head()
# Convert the dictionary into DataFrame
df = pd.DataFrame(data)
# select two columns
df[['DateTime', 'Value']]
df.plot()
pyplot.legend(loc='upper left')
pyplot.show()
```

Get data from Cloud

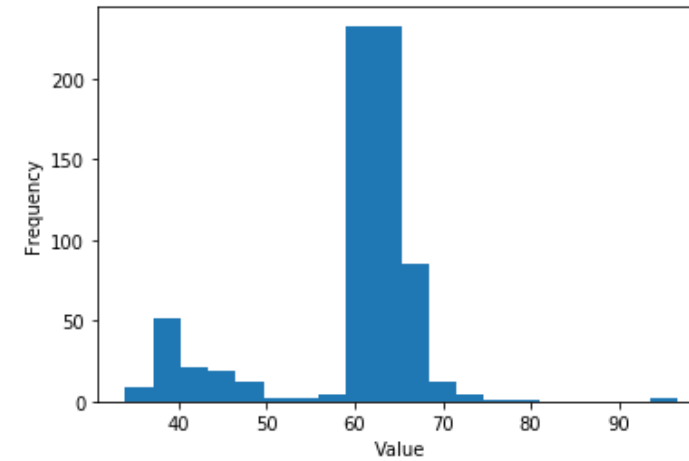


Microsoft Excel  
ma Separated Valu



```
In [6]: pyplot.hist(df["Value"],bins=20)
pyplot.xlabel("Value")
pyplot.ylabel("Frequency")
```

Out[6]: Text(0, 0.5, 'Frequency')



```
In [7]: rolling_mean = df.Value.rolling(window=12).mean()
rolling_mean2 = df.Value.rolling(window=24).mean()
pyplot.plot(df.DateTime, df.Value, label='%RH')
pyplot.plot(df.DateTime, rolling_mean, label='%RH 1 hr SMA', color='orange')
pyplot.plot(df.DateTime, rolling_mean2, label='%RH 2 hr SMA', color='red')
pyplot.legend(loc='upper left')
pyplot.show()
```

