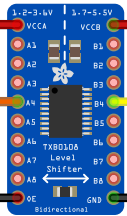


ESP32 Flow Meter

By,
Roy H Guerra Jr.

5 Volt to 3 Volt Level Converter



Uxcell G3/4" Hall Effect Flow Sensor (2-45 L/min)
1L/min = 477 pulses/min
Yellow = Pulse Out
Red = +5Vdc
Black = Gnd

Add a micro-USB 5VDC Power Adapter Here.....

Note-could use a Heltec ESP32 WiFi Module with a built in (128X64) OLED, or use a separate (128X64) OLED with IIC interface. Wiring is:

- GPIO13 to orange wire

For an External OLED Module:

- Vcc = 3.3 VDC

- GND = GND

- SCL = GPIO15

- SDA = GPIO4

```

1  # Main Server Program
2  # =====
3
4  # Import Modules
5  # -----
6  #import network
7  from time import sleep
8  import time
9  #import wifimanager # Start the Wifi Manager Program (this is a separate file)
10
11 #wlan = wifimanager.get_connection()
12 #if wlan is None:
13     #print("Could not Initialize Network Connection.")
14     #while True: # Program will not go further and stat_model() will not operate
15         #pass
16
17 # Shut off Access Point
18 # -----
19 #ap_if = network.WLAN(network.AP_IF)
20 #sleep(5) # 5 second time delay
21 #ap_if.active(False) # Turn off AP Mode
22
23
24 # Import Modules
25 # -----
26 #from network import WLAN
27 #from microWebSrv import MicroWebSrv
28 import machine
29 from machine import Pin, I2C
30 from ssd1306 import SSD1306_I2C
31 #from machine import WDT
32 #wdt = WDT(timeout=5000) # enable it with a timeout of 5s
33 #wdt.feed()
34
35 # Constants & Variables:
36 # -----
37 FLOW_PIN = 13 # Flow Sensor Input
38 OLED_RST_PIN = 16 # OLED Reset Pin
39 WIFI_LED_PIN = 25 # ON Board LED
40 int_count = 0 # Interrupt Counter
41 total_count = 0 # Pulse Totalizer variable
42 L_min = 0.0 # Calculated Flow Rate in L/min
43 gpm = 0.0 # Calculated Flow rate in gpm
44 MEASURE_DELAY = 1000
45 _lasttime = time.ticks_ms()
46
47 # Initial Setup
48 # -----
49 sensor = (Pin(FLOW_PIN, Pin.IN, Pin.PULL_UP)) # Flow Sensor PIN with pull-up resistor
50 led1 = Pin(WIFI_LED_PIN, Pin.OUT) # Sets up WiFi LED
51 OLED_RST = Pin(OLED_RST_PIN, Pin.OUT, value=1) # Sets up OLED RST to high
52 i2c = I2C(sda = Pin(4), scl = Pin(15)) # Set Up OLED IIC Pins
53 display = SSD1306_I2C(128, 64, i2c) # IIC parameters for OLED Display
54
55 # Hardware Interrupt Function
56 # -----
57 def callback(pin):
58     global int_count
59     int_count += 1 # Increment interrupt counter variable by "1"
60
61 # Set I/O pin as an Interupt Trigger on Falling Edge
62 # -----
63 sensor.irq(trigger=Pin.IRQ_FALLING, handler=callback)
64
65 # Generic Function to Display 6 lines on OLED

```

```

66 # -----
67 def Display_Stat1(line1,line2,line3,line4,line5,line6,delaytime):
68     display.fill(0) # Clear Previous Text
69     display.text(line1,0,0,1) #Line 1
70     display.text(line2,0,10,1) # Line 2
71     display.text(line3,0,20,1) # Line 3
72     display.text(line4,0,30,1) #Line 4
73     display.text(line5,0,40,1) # Line 5
74     display.text(line6,0,50,1) # Line 6
75     display.show() # Display New Text
76     sleep(delaytime) # Delay secs
77
78 # Display Start Message
79 # -----
80 L1 = "WiFi Flow Sensor"
81 L2 = "By Roy Guerra"
82 L3 = ""
83 L4 = "Starting"
84 L5 = "Please Wait"
85 L6 = ""
86 delayx = 3
87 Display_Stat1(L1,L2,L3,L4,L5,L6,delayx) # Goto Display function
88
89 # Function Check to see if Station mode wifi is active
90 # -----
91 def stat_model():
92     stal = WLAN(network.STA_IF)
93     if not stal.isconnected():
94         led1.value(0) # Turn off LED
95         L1 = "Press Reset"
96         L2 = "To Re-Start"
97         L3 = "WiFi Connection"
98         L4 = "or"
99         L5 = "Check Router"
100        L6 = "Control Settings"
101        delayx = 2
102        Display_Stat1(L1,L2,L3,L4,L5,L6,delayx) # Goto function
103    else: # connected to station wifi
104        led1.value(1) # Turn on LED
105
106 # Main Program
107 # -----
108 while True:
109     try:
110         newtime = time.ticks_ms()
111         if int_count > 0: # Check for interrupt
112             state = machine.disable_irq() # Disable interrupt, if ocurred
113             int_count -= 1 # Reset interrupt counter
114             machine.enable_irq(state) # Enable interrupt
115             total_count += 1
116             #print("Total Counts = " + str(total_count)) # Used to debug
117             #print("") # Used to debug
118         if newtime - _lasttime < 0 or newtime - _lasttime > MEASURE_DELAY:
119             state = machine.disable_irq() # Disable interrupt
120             L_min = float("{0:.2f}".format(total_count / 8.00)) # Pulses converted to
121             L/min as float
122             lph = "{0:.2f}".format(L_min*60) # Convert to a new unit and a string
123             gpm = "{0:.2f}".format(0.2641721 * L_min) # Convert to a new unit and a
124             string
125             gph = "{0:.2f}".format(L_min*15.85) # Convert to a new unit and a string
126             print("Delta Time = " + str(newtime - _lasttime))
127             print("Total Counts = " + str(total_count)) # Used to debug
128             print("LPM = " + str(L_min)) # Used to debug
129             print("LPH = " + lph) # Used to debug
130             print("GPM = " + gpm) # Used to debug
131             print("GPH = " + gph) # Used to debug
132             if L_min > 45.0:

```

```

131         print("Sensor is OverRange") # Used to debug
132         L1 = "Sensor Is"
133         L2 = ""
134         L3 = "Overrange"
135         L4 = ""
136         L5 = ""
137         L6 = ""
138         delayx = 0.01
139         Display_Stat1(L1,L2,L3,L4,L5,L6,delayx) # Goto Display function
140     else:
141         L1 = "Flow Rates Are:"
142         L2 = "-----"
143         L3 = "LPM = " + str(L_min)
144         L4 = "LPH = " + lph
145         L5 = "GPM = " + gpm
146         L6 = "GPH = " + gph
147         delayx = 0.01
148         Display_Stat1(L1,L2,L3,L4,L5,L6,delayx) # Goto Display
149         #stat_model() # Goto WiFi Function to see if connected
150         _lasttime = newtime
151         int_count = 0 # Reset interrupt counter to "0"
152         total_count = 0
153         machine.enable_irq(state) # Enable interrupt
154
155     except Exception as e:
156         print(e)
157         L1 = "Bad Connection"
158         L2 = ""
159         L3 = "and / or a"
160         L4 = ""
161         L5 = "Program Issue"
162         L6 = ""
163         delayx = 2
164         Display_Stat1(L1,L2,L3,L4,L5,L6,delayx) # Goto Display function
165
166

```