Arduino Data Logger

Jim Giammanco, N5IB
n5ib@juno.com http://www.qsl.net/n5ib

This article describes a versatile data logger based on the Arduino NANO microcontroller. Its original purpose was to record meteorological data. But it can be adapted for many other applications requiring the monitoring and recording of analog and digital data.

Pre-assembled subsystem modules are extensively used to simplify construction, as well as being cost effective. Little is required of the main circuit board other than interconnecting the various sub-modules and providing for connection to a power source.

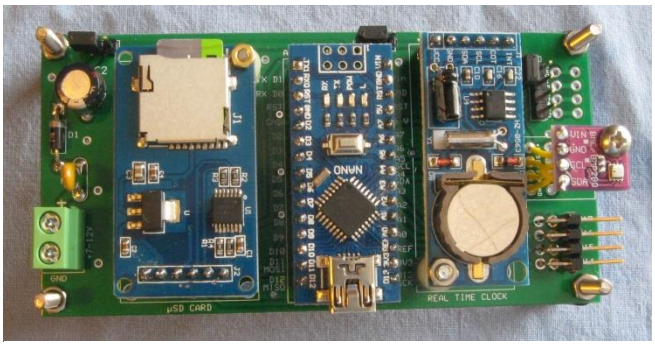


Figure 2. Data Logger modules - left to right: µSD Card reader/writer, Arduino NANO microcontroller, Real time clock/calendar, Pressure/temperature, humidity sensor.

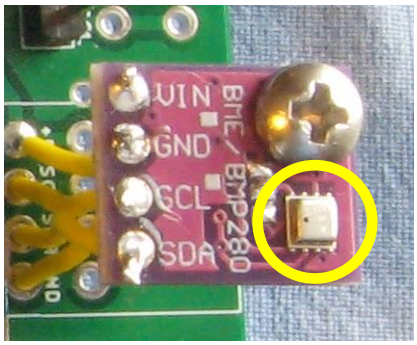


Figure 1. Close up of BME-280 Pressure, Temperature, and Humidity sensor module. The sensor chip is circled

Core System

There are four main subsystems which are generally common to any data logging device:

Control Module

An Arduino NANO microcontroller supervises the logger. It incorporates analog inputs and digital inputs/outputs, as well as communications to the peripheral modules via the SPI or I2C bus protocols, plus conventional asynchronous serial communications.

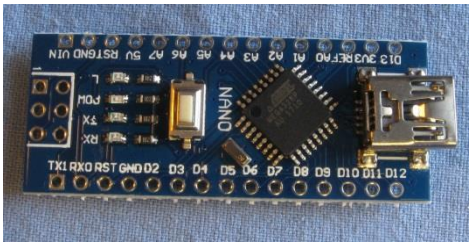


Figure 3. Arduino NANO (clone) without header pins

The microcontroller is programmed using the downloadable Arduino IDE (Integrated Development Environment) by connecting to the on-board USB port. The program source code files are downloadable from links listed in the Bill of Materials.

Data Storage Module

The logger stores its measured data onto a micro-SD memory card. The data are saved as human-readable ASCII text files in standard FAT32 file structure. The card reader/writer module used communicates with the microcontroller via the SPI bus

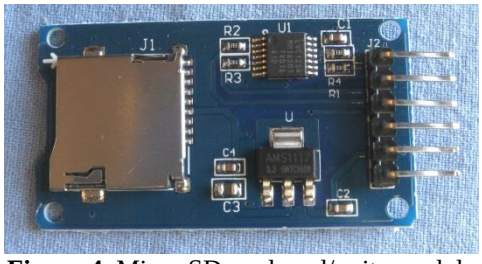


Figure 4. Micro SD card read/write module with factory installed header pins

protocol. Use a standard FAT32 formatted micro SD card. It need not be one of the very high speed cards.

Real-Time Clock/Calendar Module

To provide a date/time stamp for each saved data record, a quartz crystal-controlled, battery-backed-up real time clock module is used. It communicates with the microcontroller using the I2C bus protocol. Once set, the clock has an accuracy of a few seconds per month, and the backup battery maintains the timing even when the logger is unpowered.

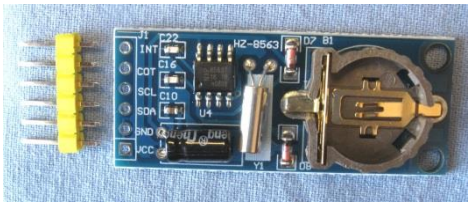


Figure 5. Real Time Clock Module with factory-supplied, but uninstalled header pins

Communication and Display

For local display of the data being monitored, an LCD display panel is used. The microcontroller communicates with the LCD via the shared I2C bus. Use of an I2C capable display saves several of the microcontroller's I/O lines for data collection. The program is written for a 2-line by 16-character panel, but larger sizes can easily be accommodated by changing just a few lines of program code.

As an additional convenience, a BlueTooth serial data module can be added. A computer, smartphone , or Android tablet can then be used to wirelessly access the data being monitored.

The display pictured here is a white-on-blue backlighted panel. The square blue trimmer on the I2C interface board is the contrast control.

Figure 6. LCD 2x16 display .
Top: front side while unpowered
Middle: front side while operational
Bottom: back side showing I2C interface board and contrast trimmer

Sensor Array

To perform its meteorological mission this logger also incorporates several environmental sensors:

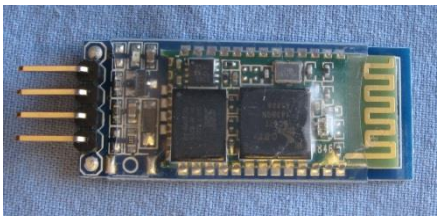


Figure 7. BlueTooth serial module with factory-installed header pins

Thermometer, Hygrometer (Relative Humidity), and Barometric Pressure

The Bosch BME-280 sensor incorporates a temperature sensor and temperature-compensated sensors for relative humidity and barometric pressure. It communicates with the microcontroller via the shared I2C bus.

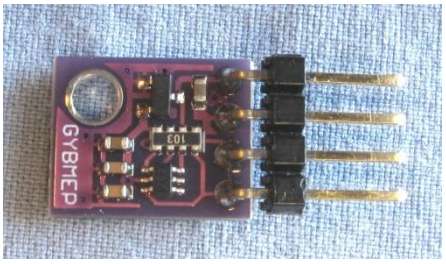


Figure 8. Bosch BME-280 sensor module with user installed header pins

The software library that supports the BME-280 includes functions to select the desired units of measurement, as well as a function to calculate the dew point based upon the temperature and humidity measurements.

Rain Gauge

A *tipping bucket* rain gauge is used to measure liquid precipitation. This type of rain gauge is self-emptying and can measure rainfall with a resolution down to 0.01 inches. The particular unit used with the prototype logger was salvaged from an old consumer weather station, and had a resolution of 0.02 inches.

A funnel channels collected rainfall onto a dual-chamber *bucket* that tips back and forth and empties itself when it has accumulated a measured quantity of water. After a tip that empties one side of the bucket the other side is then in position to collect, and the cycle repeats. Each time the bucket tips a magnet passes over a magnetic reed switch and causes a momentary contact closure, which is sensed by a digital input on the microcontroller module. Counting the number of tip events yields the accumulated rainfall. Additionally, measuring the time between successive tip events allows a calculation of the rainfall rate.

There were several 3-D printable designs found on the internet. A search for "3D printable tipping bucket rain gauge" turned up this link among many others:
https://www.thingiverse.com/thing:2757560

System Housekeeping

Power Source

The logger can be powered with a +5 V DC source connected via the USB connection on the Arduino NANO module. Alternatively, screw terminals are provided to attach an external DC source of from 7 to 12 V. Current demand is very modest: less than 50 mA with BlueTooth and Backlighted LCD options installed.

Configuration jumpers

Three configuration jumpers are used to select the power source (JP1), turn the LCD panel ON/OFF (JP2), and turn the BlueTooth ON/OFF (JP3). In addition, a switch or shorting jumper can be connected to the LCD's I2C interface board to turn the LCD backlight ON/OFF while leaving the display powered ON. It is even possible to connect a variable resistor instead of a switch or jumper, to allow a backlight brightness adjustment. Solder-bridge jumper (SJ1) is only closed if the system needs to accept incoming data from the optional BlueTooth module. It must be left open (or the BlueTooth module removed) when uploading programs to the Arduino.

Additional Inputs and Outputs

Only a few of the I/O lines available on the Arduino NANO are used for the basic logger functions. The unused digital and analog lines are brought out to PCB pads or to header pins to allow for additional data input.

There are extra sets of pads for additional I2C bus connections – one set at the BME-280 location, and two sets at the LCD location. Header strips can be installed there to allow additional I2C devices to be connected if desired. One possibility is an extra BME-280 to allow indoor and outdoor measurements.

Normally the ADC (analog to digital converter) on the Arduino NANO uses the 5V power supply as its voltage reference. But provisions are made on the main circuit board to install a precision band-gap voltage reference if desired. CAUTION: Should this be done, the program source code MUST be modified to specify use of an EXTERNAL reference. Otherwise the Arduino's voltage reference input will be shorted to the 5 V supply and could be damaged.

Programming

Programs (called *sketches* in Arduino-ese) are uploaded from a host PC into the Arduino NANO via the USB connection. Consult the information and tutorials for the Arduino *Integrated Development Environment* (IDE) for the details of the editing, compiling, and uploading process as well as the procedure for loading the specialized libraries. The IDE can be obtained – a free download – from *https://www.arduino.cc/en/main/software*.

Note: *If the BlueTooth serial module is installed, either it must be temporarily removed, or solder jumper SJ-1 opened, while uploading sketches to the Arduino. Otherwise the BlueTooth module's data transmit line will interfere the Arduino receiving the data stream from the host computer.*

The documentation package for the data logger contains the source code for the main logger sketch, as well as several short sketches used to test and verify the several submodules during construction. It also contains copies of the specialized program libraries that are needed to support the submodules.

Main Logger Sketch (*BME280_BT_LCD_RTC_EEPROM_Wx_Logger3r0.ino*)

I2C bus address scanner (*I2C_Scanner_N5IB.ino*)

Real time clock setting and verifying (*set_8563_clock_N5IB.ino*)

Demonstration clock (*Demo_clock_N5IB.ino*)

Micro SD card read/write testing (*SPI_uSD_Tester.ino*)

LCD display tester (*HelloWorld_i2c_LCD_N5IB.ino*)

BME-280 tester (*Example1_BasicReadings_N5IB.ino*)

EEPROM write/read tester (*EEPROM_write_and_read_tester.ino*)

Libraries

Real-time Clock Library *(Rtc_Pcf8563-master.zip*)
LCD display Library (*Newliquidcrystal_1.3.5.zip*)
BME-280 sensor library (*SparkFun_BME280_Arduino_Library-master.zip*)
SD Card Library (already included within the Arduino IDE distribution)

Construction

Since several pre-assembled submodules are used, assembly of the main circuit board is straightforward. There are fewer than two dozen components to be installed. A few of these are surface mount (SMD) capacitors and resistors which will require a fine tipped soldering iron and small diameter solder. The component locations are identified on the top and bottom screen printed labels on the board. Note that the SMD components are installed on the bottom side, while the through-hole components are mounted from the top side.

One issue to be addressed during construction will be the choice of connectors to be used for attaching the several submodules. Connector orientation (straight or right-angled) and connector gender are at the discretion of the builder. The most compact arrangement (as shown in the earlier photograph) will result when vertical header sockets are mounted on the main board and vertical header pins are used on the submodules.

Wherever possible, the submodules should be purchased from vendors who will supply them ***without*** header pins installed. This has been possible for the Arduino NANO, Real Time Clock, and BME-280 modules. To date, a source for the BlueTooth and Micro SD modules without pins hasn't been found. The builder will have to decide whether to utilize main board connectors that match the existing pins on the modules, or else remove the existing pins and replace them, as was done for the unit in the photograph.

Operation

Be sure a fresh battery is installed on the real time clock module. Then compile, upload, and run the clock-setting sketch.

Next compile, upload, and run the EEPROM test sketch. Use it to initialize the rain gauge counter. If you know how much rainfall you have accumulated year-to-date you can calculate the starting counter value:

Integer counter value = inches of rainfall year-to-date ÷ 0.02

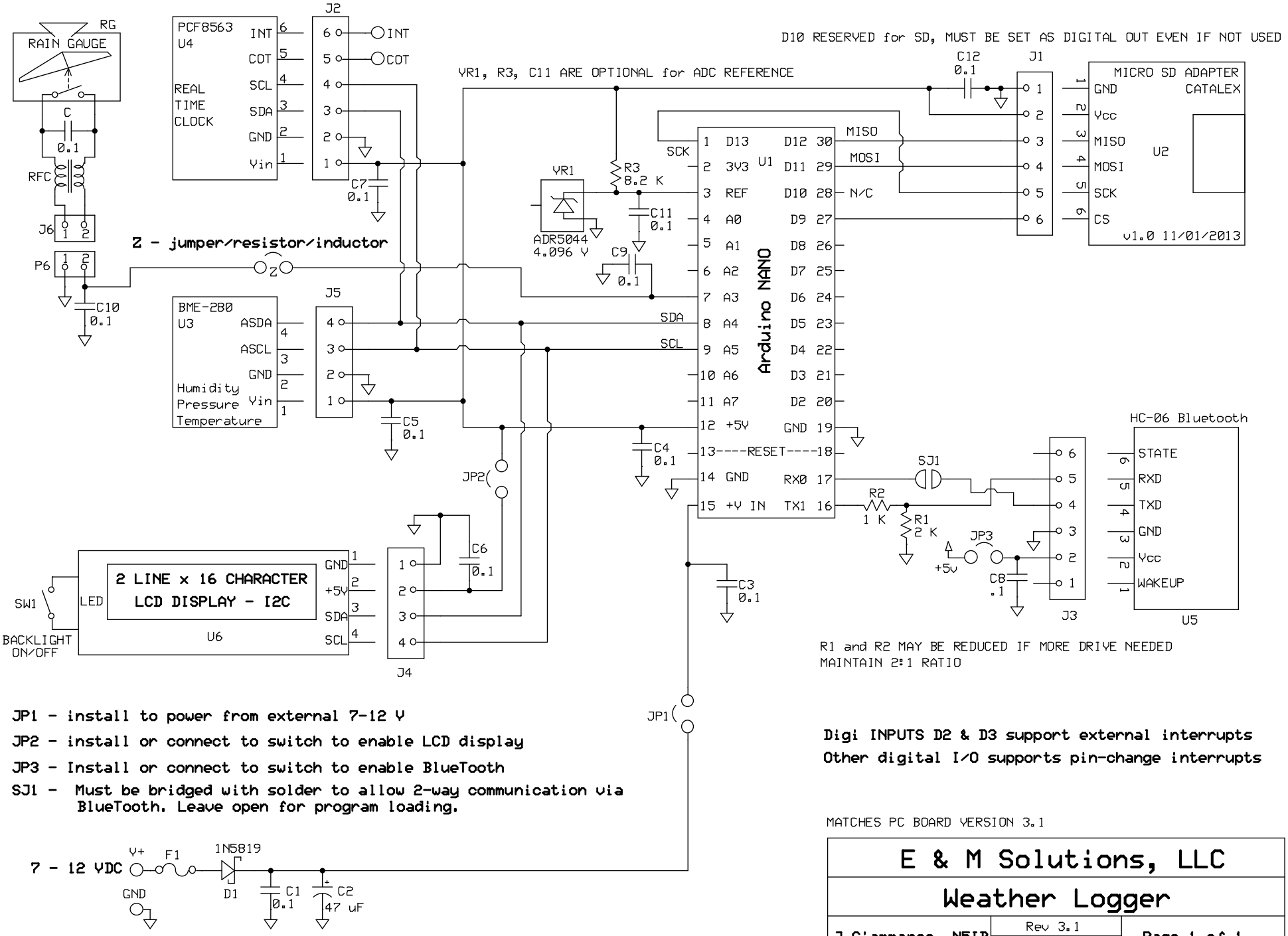
Otherwise the counter value should be initialized to zero. This integer value (round it off if there's a fraction) should be written into EEPROM address zero.

Finally insert a formatted micro-SD card into the SD module and compile, upload, and run the main logger sketch. The sketch will save the data in an ASCII text file whose name will be in the form:

mmmyyyy.txt

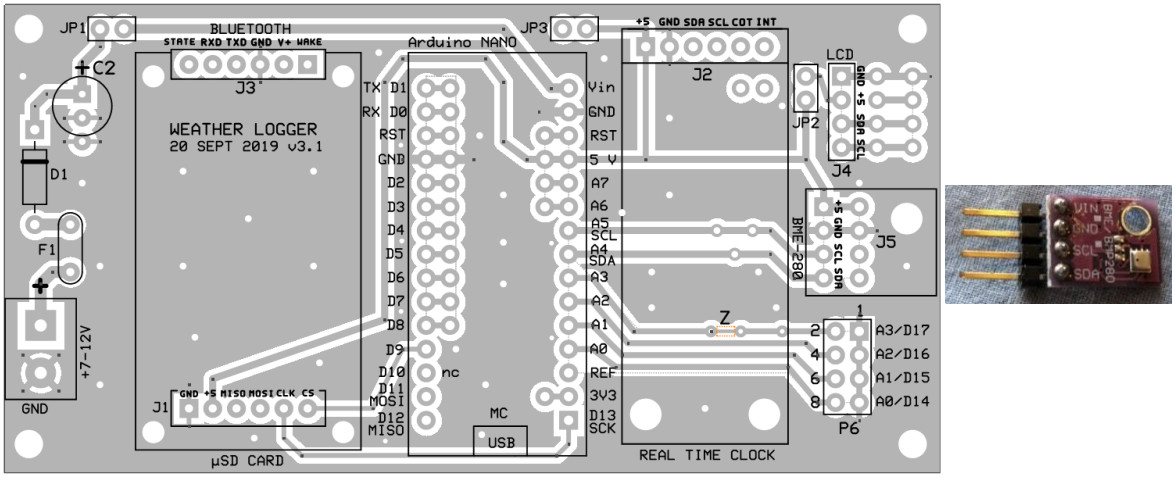
Where *mmm* is the three character abbreviation of the current month; and *yyyy* is the current year, as retrieved from the real time clock module.

The LCD display should briefly show each of the three *splash screens* below before reverting to a periodically updating data display. The parameters displayed are the temperature, barometric pressure, relative humidity, accumulated rainfall, and the rate of rainfall if there has been a recent rain gauge tip event.

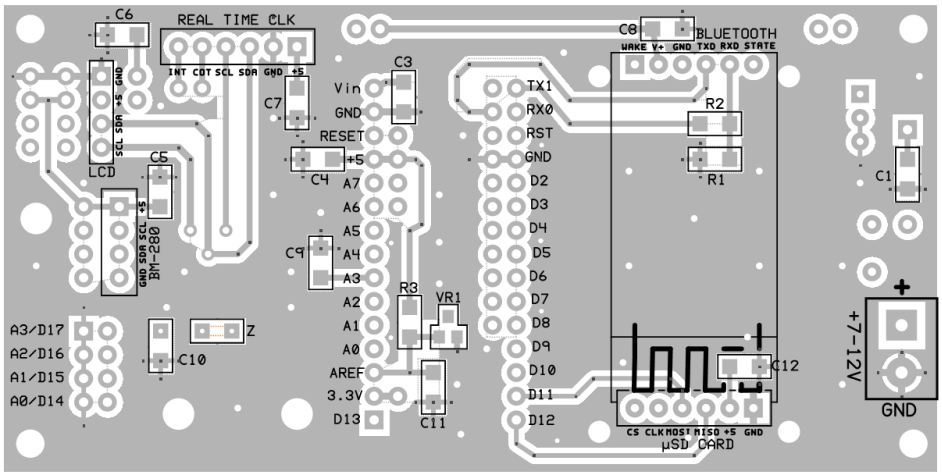

RG
RAIN GAUGE
C
0.1
RFC
J6
P6
C10
0.1
PCF8563
U4
REAL
TIME
CLOCK
INT
COT
SCL
SDA
GND
Vin
J2
C7
0.1
Z - jumper/resistor/inductor
BME-280
U3
Humidity
Pressure
Temperature
ASDA
ASCL
J5
C5
0.1
D10 RESERVED for SD, MUST BE SET AS DIGITAL OUT EVEN IF NOT USED
YR1, R3, C11 ARE OPTIONAL for ADC REFERENCE
VR1
ADR5044
4.096 V
R3
8.2 K
C11
0.1
C9
0.1
C12
0.1
J1
MICRO SD ADAPTER
CATALEX
GND
Vcc
MISO
MOSI
SCK
CS
U2
v1.0 11/01/2013
SCK
MISO
MOSI
N/C
SDA
SCL
Arduino NANO
U1
D13
3V3
REF
A0
A1
A2
A3
A4
A5
A6
A7
+5V
RESET
GND
+V IN
D12
D11
D10
D9
D8
D7
D6
D5
D4
D3
D2
GND
RX0
TX1
C4
0.1
JP2
C6
0.1
C3
0.1
SJ1
R2
1 K
R1
2 K
JP3
+5v
C8
.1
J3
HC-06 Bluetooth
STATE
RXD
TXD
GND
Vcc
WAKEUP
U5
SW1
BACKLIGHT
ON/OFF
LED
2 LINE x 16 CHARACTER
LCD DISPLAY - I2C
U6
GND
+5V
SDA
SCL
J4
R1 and R2 MAY BE REDUCED IF MORE DRIVE NEEDED
MAINTAIN 2:1 RATIO
JP1
JP1 - install to power from external 7-12 V
JP2 - install or connect to switch to enable LCD display
JP3 - Install or connect to switch to enable BlueTooth
SJ1 - Must be bridged with solder to allow 2-way communication via BlueTooth. Leave open for program loading.
Digi INPUTS D2 & D3 support external interrupts
Other digital I/O supports pin-change interrupts
7 - 12 VDC
V+
GND
F1
1N5819
D1
C1
0.1
C2
47 uF
MATCHES PC BOARD VERSION 3.1
E & M Solutions, LLC
Weather Logger
J Giammanco, N5IB
Rev 3.1
15 OCT 2019
Page 1 of 1

Printed Circuit Board Images
Images are not actual size
Actual dimensions are 10 cm wide by 50 cm high

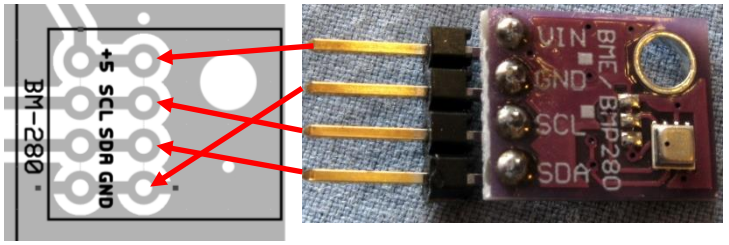
Top Copper v3.1



Bottom Copper v3.1

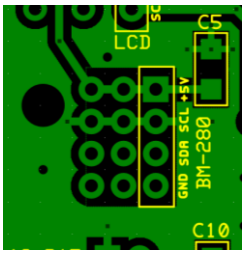


******* Errata – Version 3.0 PCB only *******

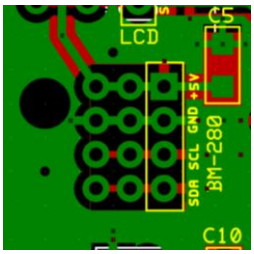


There is an error on the prototype v3.0 PC board at the location of the header strip that receives the BME-280 module. The photos at right show the v3.0 PCB header location and the actual BME-280 module pins. Individual wire connections (red arrows) will be needed to attach the BME-280. The screen printed labels on the PCB do correctly identify the pins that must be connected to the correspondingly labelled pins of the BME-280. Note (top photo) that on the newer version 3.1 and 3.2 PCBs the pins are correctly arranged to directly receive the BME-280 module. Use the set of four pins farthest from the board edge for the on-board BME-280 module. The second (and third on v3.2) set of pins can be used for additional (possible remote) modules, or even some other, unrelated I2C devices.

******* Errata – Version 3.1 PCB only *******



On the v3.1 PCB there is an error in the bottom-side silkscreen label for the BME-280 sensor module header pins. The labels were a relic of the v3.0 board layout. The image at left shows the ***incorrectly*** printed label. Note the misplaced labels for the lower three pins.
At right is a ***correct*** label as it should appear.



The top-side silkscreen label is correct as printed. And all of the copper interconnections are correct. Both silkscreen labels are correct on the newest PCB, v3.2

Bill of Materials for v3.0, v3.1, and v3.2 PC boards

C1, C3, C4, C5, C6, C7 C7, C9, C10, C12	0.1 µF 50V SMD 1206 ceramic capacitor
C2	47 uF or greater, 25 V or greater, aluminum electrolytic capacitor, 0.1" or 0.2" lead spacing
D1	1N5819 or 1N5818 Schottky rectifier
F1	Resettable fuse, 100 mA hold current, 0.2" lead spacing – optionally, omit and use wire jumper
J1, J2, J3	6-pin vertical female header strip – builder's choice of gender, vertical, or right angle to match modules
J4, J5	4-pin header strip – builder's choice of gender, vertical, or right angle to match modules
J6	8 pin (2 x 4) header – builder's choice of gender, vertical, or right angle
P6	mating connector to match J6 - – builder's choice of gender, vertical, or right angle
JP1, JP2, JP3	2-pin vertical male headers with shorting shunts, or optionally, wire jumpers

U1	Arduino (or clone) NANO microcontroller module, pre-assembled (example: eBay item # 162933350570)
U2	Catalex µSD Card Read/Write module, pre-assembled (example: eBay item # 391720940701)
U3	BME-280 Pressure, Temperature, Humidity module, pre-assembled (example: Amazon ASIN # B07KYJNFMD)
U4	PCF8563 Real Time Clock module, pre-assembled (example: eBay item # 362714293867)
Battery	CR1220 3V Lithium watch battery for clock backup (a CR1216 can be used in a pinch if CR1220 not available)

U5	HC-06 Bluetooth Serial Module, pre-assembled (example: eBay item # 323326734652)		)
R1	2 K	5% SMD 1206	) optional
R2	1 K	5% SMD 1206	)
C8	0.1 µF 50V SMD 1206		)

U6	LCD Display, I2C interface, minimum size 2 lines by 16 characters (example: Amazon ASIN # B0711WLVP9)
SW1	optional SPST switch to enable/disable LCD display backlight

RG	Tipping Bucket Rain Gauge, momentary contact closure on each tip event
C	0.1 µF capacitor across rain gauge contacts
RFC	bifilar RF choke in series with rain gauge leads – 6 to 10 turns bifilar on an FT37-43 or FT50-43 toroid
Z	shorting jumper, optionally RF choke or low value resistor

VR1	ADR5044WARTZ-R7 4.096V precision shunt voltage reference		)
R3	8.2 K	5% SMD 1206	) optional
C11	0.1 µF 50V SMD 1206		)

Screw Terminals	2-postion terminal block 0.2" (or 5 mm) pitch (example: Marlin P Jones # 18318 TS)
PCB	printed circuit board
Hardware	mounting screw holes for the µSD adapter fit #2-56 screws, all others fit #4-40 or 3mm

Program source code files are found in a compressed (zipped) folder at this link:
https://qsl.net/n5ib/