/\*

SD card datalogger started using

https://www.arduino.cc/en/Tutorial/Datalogger

\*/

/\*

\*\*\*kernel connections

SD card reader

mega SD card reader

5V 5V

Gnd Gnd

53 SDSC

51 MOSI

52 SCK

50 MISO

TOD DS3231

5V 5V

Gnd Gnd

21 SCL

20 SDA

Bluetooth

5V 5V

Gnd Gnd

18(TX1) RxD through divider 1.2K/2.2K for 3.3V level

19(RX1) TxD

LCD removed

LCD display 5110

3.3V 3.3V

Gnd Gnd

Gnd BL

all signals through 5V/3.3V adapters

52(SCK) SCLK

51(MOSI) DIN

49 DC

48 CE/CS

47 RST

\*\*\*sensors connections

BMP280 (temperature/pressure/altitude)

SCL(level shifter) SCL

SDA(level shifter) SDA

GND GND

3.3V Vcc

DHT22 (temperature/humidity)

pinout seen from top(grid side), left to right 1 to 4

5V 1=Vcc

49 2=data connect 10K 1 to 2

3=NC

GND 4=GND

BH1750 (light)

5V Vcc

GND GND

SCL SCL

SDA SDA

GND ADDR

\*/

//look at this for data logger

//https://edwardmallon.wordpress.com/2015/12/22/arduino-uno-based-data-logger-with-no-soldering/

// Hardware SPI (faster, but must use certain hardware pins):

// SCK is LCD serial clock (SCLK) -

// MOSI is LCD DIN -

// pin 49 - Data/Command select (D/C)

// pin 48 - LCD chip select (CS)

// pin 47 - LCD reset (RST)

// Note with hardware SPI MISO and SS pins aren't used but will still be read

// and written to during SPI transfer. Be careful sharing these pins!

// conflict 5110 and SD documented here: https://forum.arduino.cc/index.php?topic=53843.0

//Adafruit\_PCD8544 display = Adafruit\_PCD8544(49, 48, 47);

//process to receive data from bluetooth

//start/stop data logging, then xmit with blue command

//pair rw\_datalogger with android device with installed Bletooth Terminal HC05

//receive data as ASCII (top right screnn)

//send log file to drive (google drive) then download to laptop

//make received file a xxx.csv file

//then load this xxx.csv file in yyy.xls file, this will regenerate the original xxx.csv file written to SD card

//some lines genertaed by BT tool need be removed