VIRTUAL COURSE
BUILD YOUR OWN DATA LOGGER
MODULE 3 - SUBMODULE 1 - LAB 1

THE ARDUINO COMMAND LINE
cmdArduino Library

What is it?

- A library that allows you write commands (functions) to control and test your device application in real time through the Arduino serial monitor.
cmdArduino Library

```cpp
#include "cmdArduino.h"
#include "DHT.h"

DHT dht(A0, DHT11);

void setup() {
  // put your setup code here, to run once:
  dht.begin();
  cmd.begin(57600);
  Serial.println("Lab 2b - Command Temp and Humidity");
  cmd.add("temp", cmdTemperature);
  cmd.add("humid", cmdHumidity);
}

void loop() {
  // put your main code here, to run repeatedly:
  cmd.poll();
}

void cmdTemperature(int argc, char **argv) {
  float temperature = dht.readTemperature();
  Serial.print("Temperature: ");
  Serial.print(temperature);
  Serial.println(" deg Celsius");
}
```

Lab 2b - Command Temp and Humidity

CMD: Command not recognized.

************** CMD **************
CMD >> temp
Temperature: 24.00 deg Celsius

************** CMD **************
CMD >> humid
Humidity: 51.00 %RH

************** CMD **************
CMD >>
cmdArduino Library

Why use it?

• Test easily and thoroughly
• Change parameters in our code, test and observe device behavior in real-time
• Saves time
• Check installed libraries work
• Useful for debugging in the field
• Useful to collect data when in the field without having to remove the SD card
• Remove before deployment: takes memory, device asleep anyway
Lab 1 Goals

• Install cmdArduino library via Arduino library manager
• Understand library functions we need to use
• Write and test our first command: ‘Hello Command Line’
Terms

- **command keyword** = the first word we type into the serial monitor, calls the command function.

- **command function** = the function that is called when we type our command keyword.

- **command list** = a list of all the command keywords, and their functions

- We define both the command keyword, and command function
Arduino Program Structure

```c
void setup() {
  // put your setup code here, to run once:
}

void loop() {
  // put your main code here, to run repeatedly:
}
```
cmdArduino - Program Structure

#include <cmdArduino.h>

Void setup()
{

  cmd.begin(baudrate)
  • Initializes serial port and structures needed by the command line.
  • Similar to Serial.begin(baudrate).

  cmd.add(cmdName, cmdFunctionName)
  • adds a new command to the command list and maps it to the function we specify
  • the command list is the list of command functions we create

  eg. cmd.add('hello', cmdHello)
  • maps the command keyword 'hello' to the function cmdHello(int argCnt, char **args)

}
cmdArduino - Program Structure

void loop ()
{
  cmd.poll()
  • Checks the serial port to see if a command keyword has been typed into the serial monitor eg 'hello'
  • Check checks if keyword matches a command specified in the command list.
  • If there’s a match, cmd.poll() calls the function specified for that command keyword
  • Case sensitive! eg. ‘hello’ typed into the serial console must have a lowercase h if that’s what we’ve specified in the cmd.add() function
}
void cmdFunctionName(int argCnt, char **args)
{
    • new function (outside of the loop function)
    • Replace cmdFunctionName with the function name specified in cmd.add()
    • Function declaration MUST include two arguments: int argCnt, char **args
    • Includes function code eg. Serial.println("Hello Command Line")
}

Our command functions always need to have two arguments: int argCnt, char **args - don’t change these!
Installing cmdArduino Library
Installing cmdArduino Library

A small command-line library to make development easier. cmdArduino allows you to interactively call functions from a serial command line for fast testing and debugging.
Writing our first command – Hello World

```c
#include <cmdArduino.h>

void setup() {
  cmd.begin(57600);
  cmd.add("hello", cmdHello);
}

void loop() {
  // put your main code here, to run repeatedly:
  cmd.poll();
}

void cmdHello(int argCnt, char **args) {
  Serial.println("Hello Command Line");
}
```
COMING UP - MODULE 3 - SUBMODULE 1

LAB 2: SETTING AND ACCESSING PARAMETERS