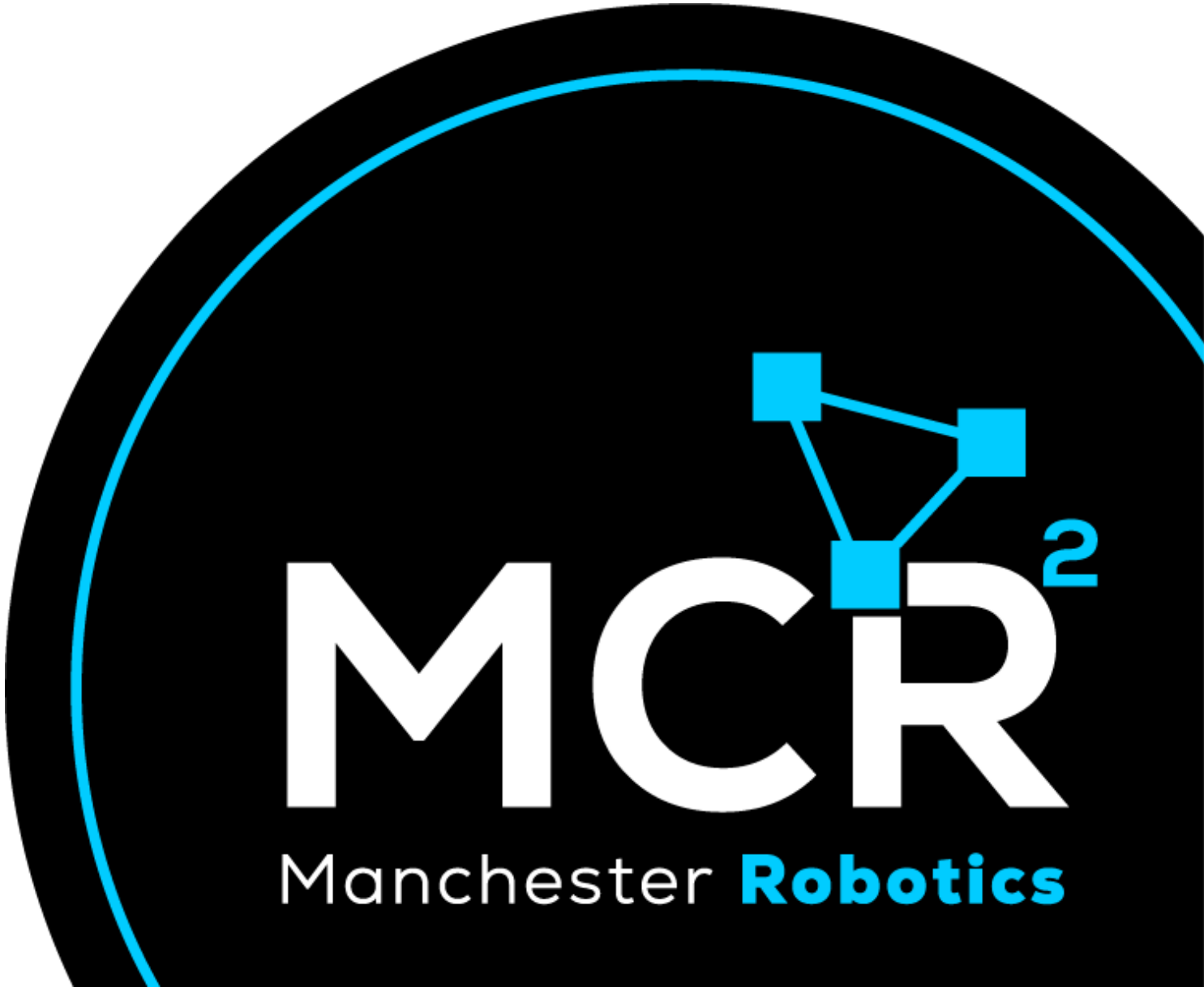
{Learn, Create, Innovate};

Standalone Programming

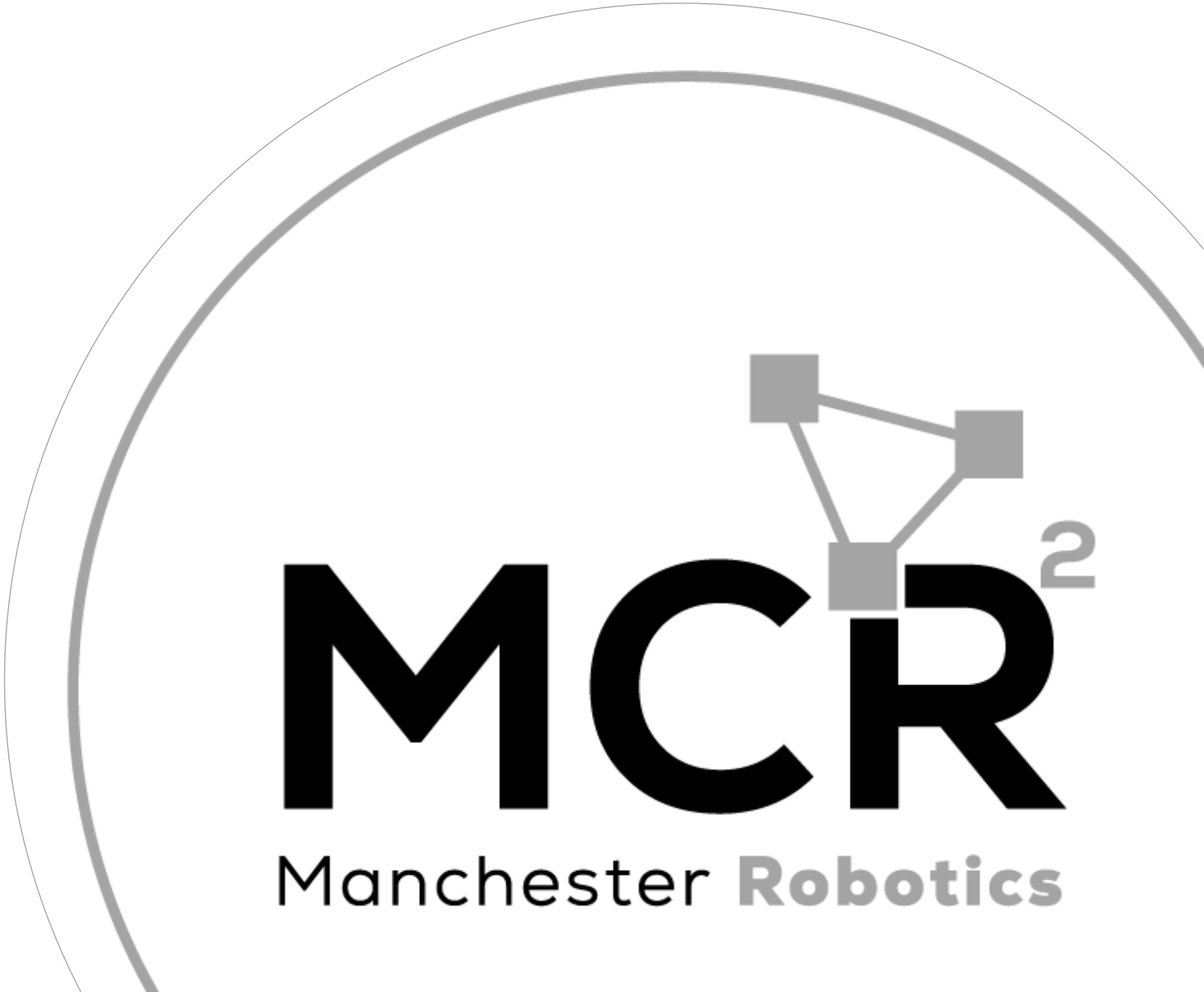
Interfacing a microcontroller and Puzzlebot



Standalone Programming

MCU Program

{Learn, Create, Innovate};



MCU Programming

General information

- Arduino and ESP32 are some of the most used MCU's.
- Both can be programmed using the Arduino IDE.
- To program the Puzzlebot, the Arduino IDE will be used.
- The MCR2 Libraries are designed to be use with the Hackerboard, the Arduino Uno and the Arduino Mega.
- Read the documentation for the libraries, on how to use it with different microcontrollers.

Arduino IDE

- An IDE, or Integrated Development Environment, helps programmers' productivity by combining common activities of writing software into a single application: editing source code, building executables, and debugging.
- Arduino IDE supports C and C++ programming languages.
- A sketch is a program written with the Arduino IDE.
- Sketches are saved on the development computer as text files with the file extension .ino.

MCU Programming

Sketch

- The simplest syntaxis for writing a sketch consists of only two functions:
- setup(): This function is called once when a sketch starts after power-up or reset. It is used to initialize variables, input and output pin modes, and other libraries needed in the sketch. It is analogous to the function main().
- loop(): The loop() function is executed repeatedly in the main program after the setup() function. It controls the board until the board is powered off or is reset.

Sketch Structure

```
// Variable declaration section


// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH);   // turn the LED on (HIGH is the voltage level)
  delay(1000);                       // wait for a second
  digitalWrite(LED_BUILTIN, LOW);    // turn the LED off by making the voltage LOW
  delay(1000);                       // wait for a second
}
```

Variable Declaration:
Libraries, Components, Variables, constants, Definitions, etc.

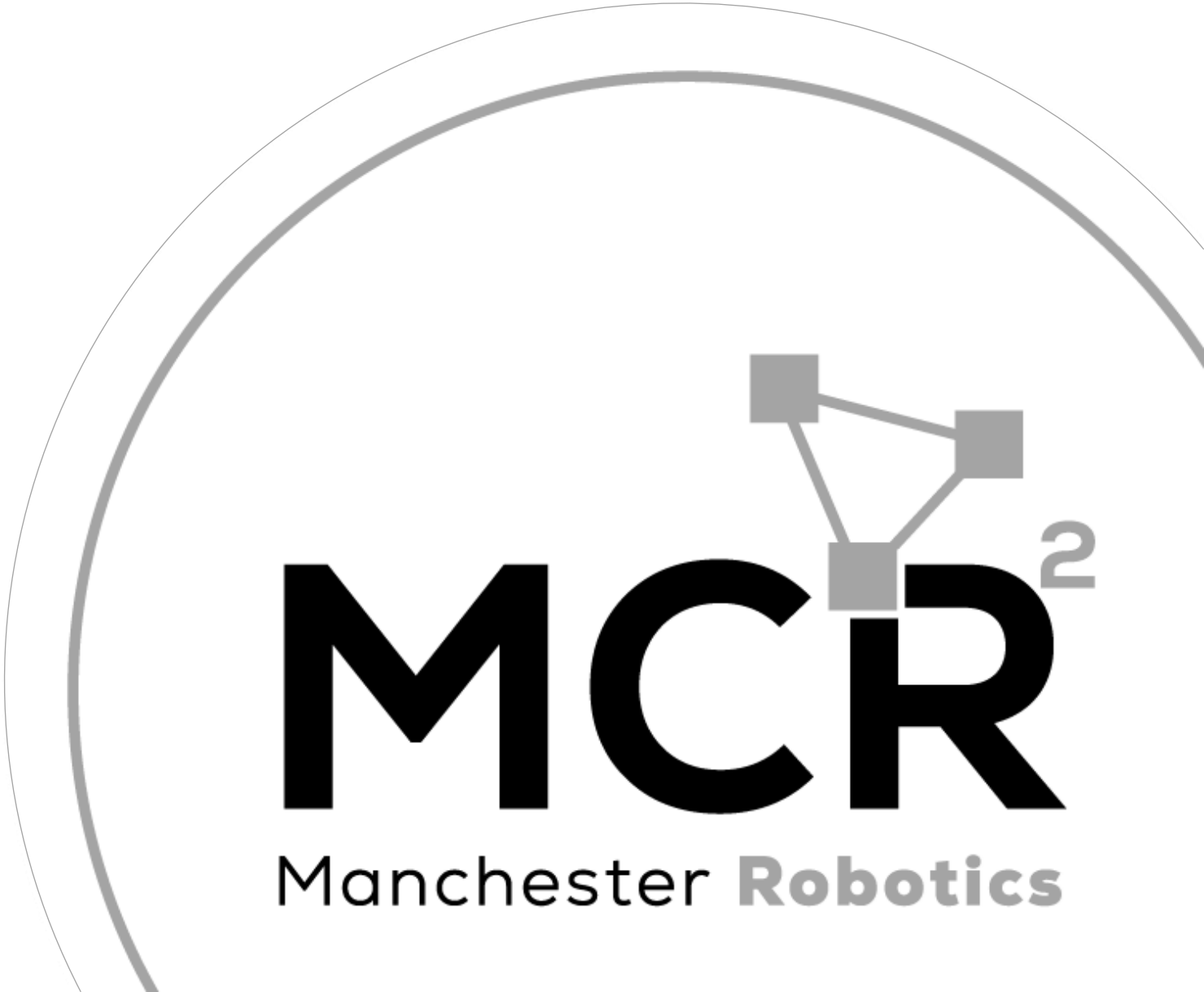
Setup Section:
Set up sensors, variables, Ports, Functions, Serial comms.

Loop Section:
Loops and repeats actions.

Standalone Programming

Example

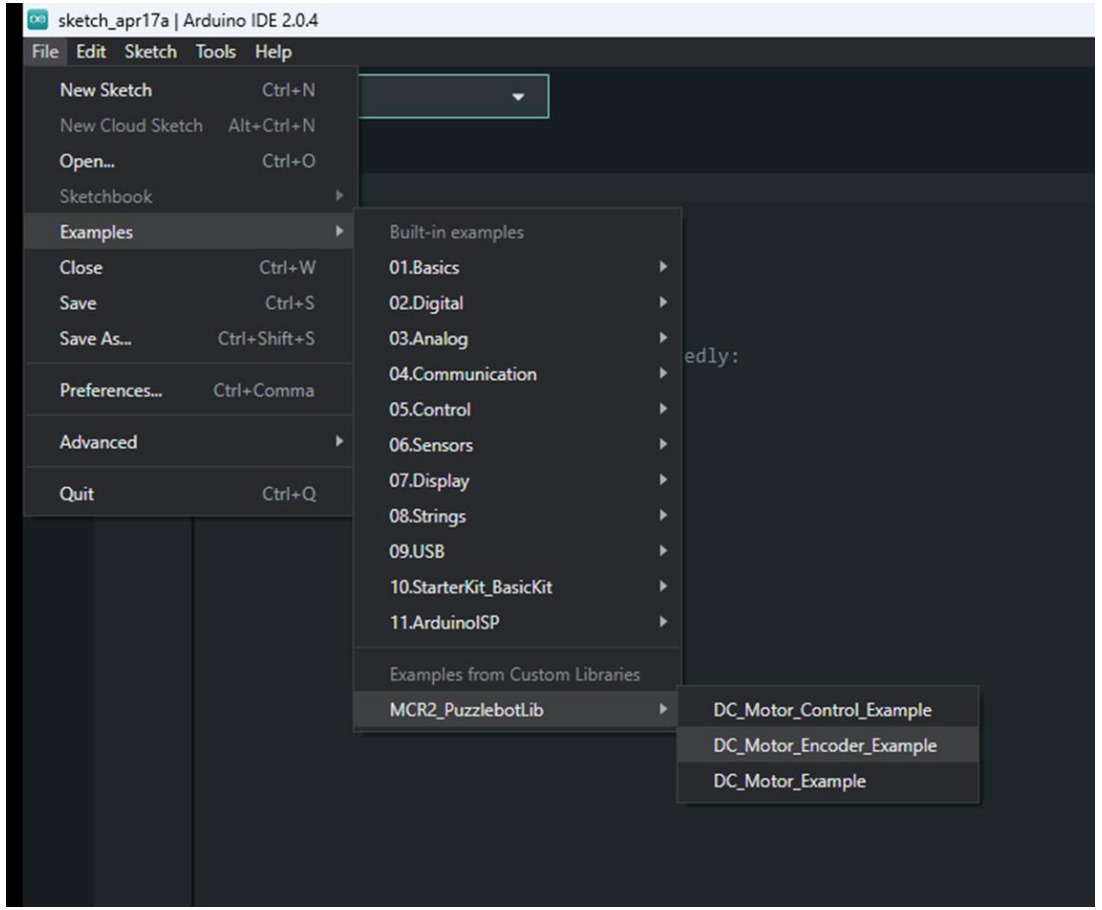
{Learn, Create, Innovate};



ROS Sketch Structure

ROS Libraries Arduino

- Look at the examples provided in the Puzzlebot libraries File > Examples > MCR2_PuzzlebotLib
- Select any example
- For this case we will use the DC_Motor_Example



ROS Sketch Structure

ROS Libraries Arduino

- Look at the examples provided in the puzzlebot libraries File > Examples > MCR2_PuzzlebotLib
- Open the Example *DC_motor_Example*
  - *This example is configured to be used with the Hackerboard.*
  - *To use it with Arduino Mega or Arduino Uno, redefine the Pins* `PWMpin, pinA, pinB`
  - *To use multiple motors with the ESP32 (Hackerboard), specify a different PWM Channel as follows*

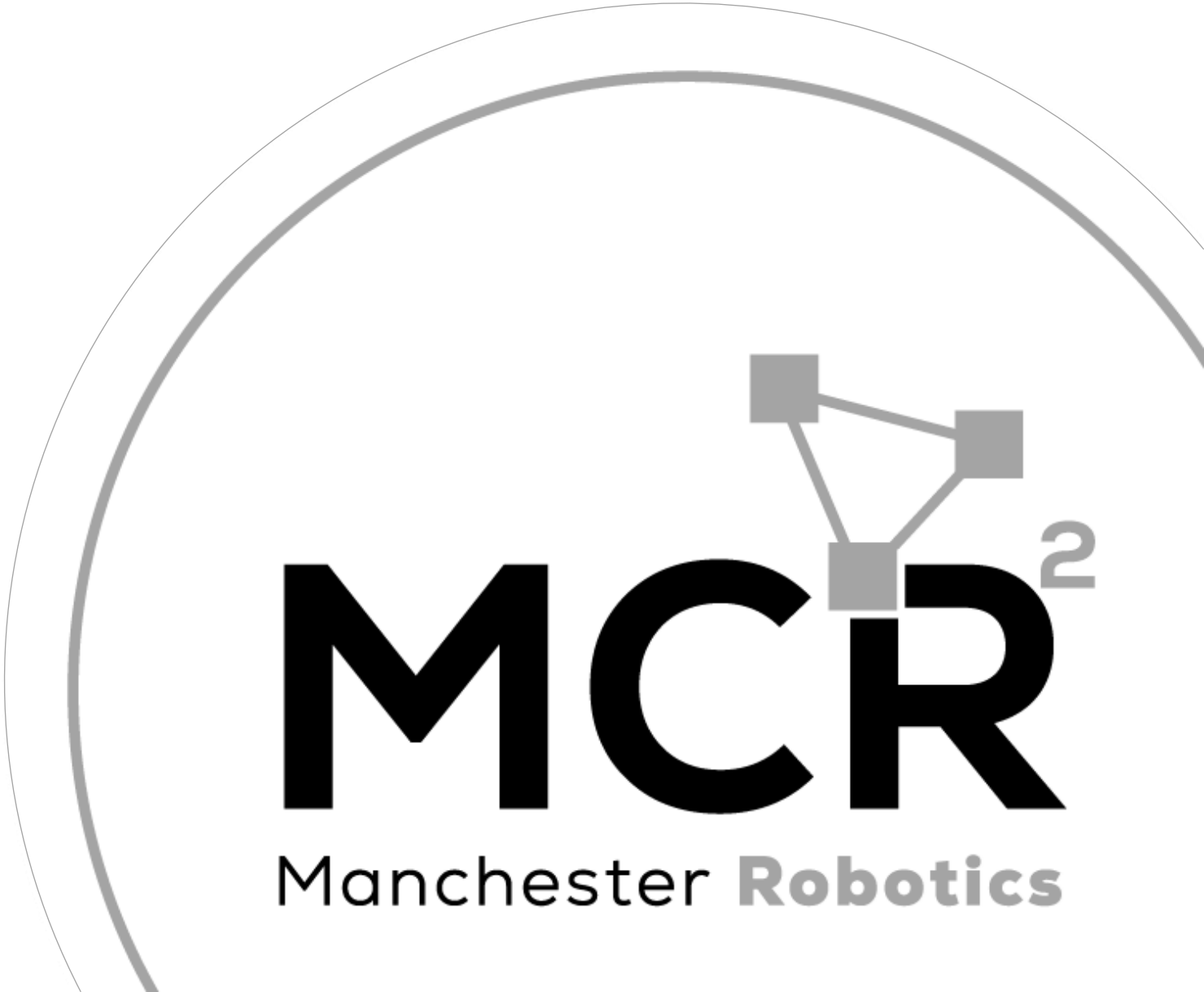
```
motor.DriverSetup(PWMpin, Channel, pinA, pinB);

motor_R.DriverSetup(motR_pins[0], 0, motR_pins[1], motR_pins[2]);

motor_L.DriverSetup(motR_pins[0], 1, motR_pins[1], motR_pins[2]);
```

```
/**

\brief Define the MotorDriver Pins and rotation
sign (PWMpin, Pin A, Pin B, Sign (-1,1))

Arduino Pins: 2,3,4 / ESP32 pins 4,15,18

*/
#ifdef ESP32
  #define PWMpin 4
  #define pinA   15
  #define pinB   18
  #define motorSign -1
#else
  #define PWMpin 2
  #define pinA   3
  #define pinB   4
  #define motorSign -1
#endif
```

ROS Serial Communication

Compilation and Uploading

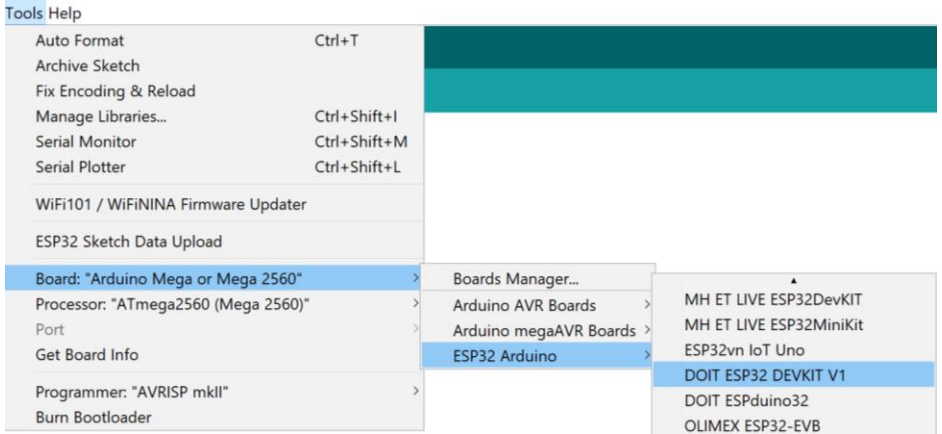
{Learn, Create, Innovate};

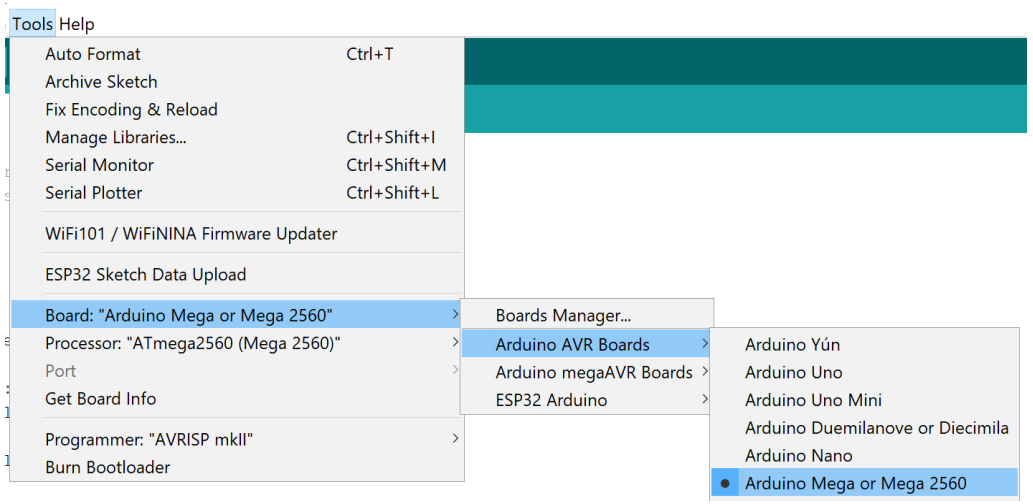


Activity

Compilation (Arduino IDE)

- Open Arduino IDE (previously configured).
- Type the code in the previous slide.
- Select the board to be used Tools>Board ESP32 for Hackeboard or Arduino Mega
  - For Arduino Select Arduino AVR Boards>Arduino Mega or Mega 2560
  - For Hackerboard select ESP32 Arduino > DOIT ESP32 DEVKIT V1





- Compile the code using by clicking check mark button located on the upper left corner.

- The following message should be displayed:

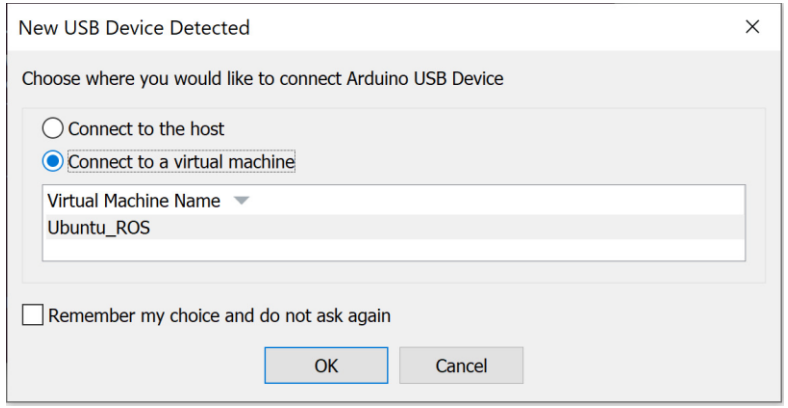
```
Done compiling.

Sketch uses 9424 bytes (3%) of program storage space. Maximum is 253952 bytes.
Global variables use 1826 bytes (22%) of dynamic memory, leaving 6366 bytes for local variables. Maximum is 8192 bytes.
```

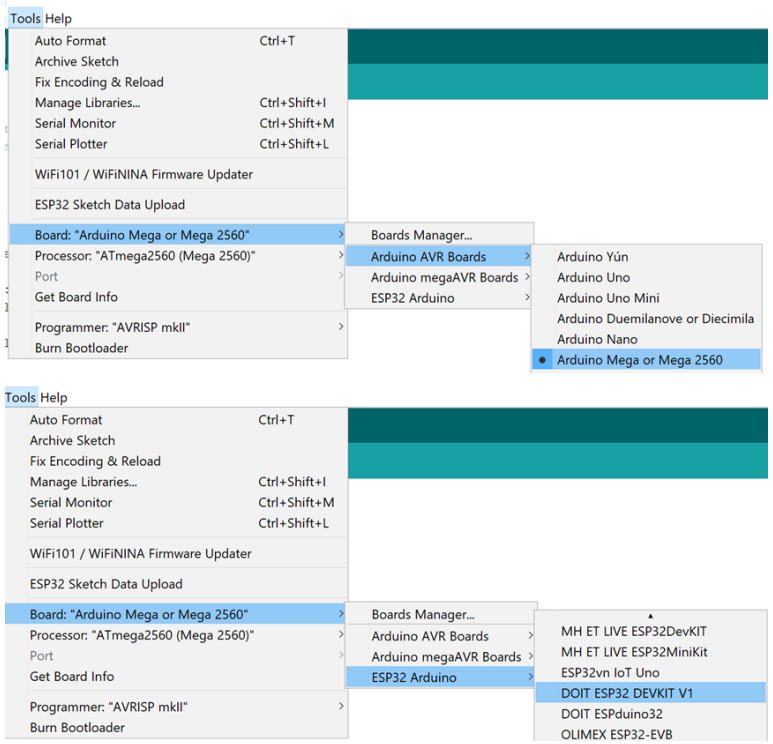
- For compilation errors or troubleshoot with the libraries, see presentation MCR2_ROS_Arduino_IDE_Configuration.

Activity

Uploading (Arduino IDE)

- Connect the board
- Select the port to be used Tools>Port
  - If working on the VM, you must first select the option Connect to a virtual machine when automatically prompted (shown) and then select the port.
- Select the board to be used Tools>Board
  - For Arduino Select Arduino AVR Boards>Arduino Mega or Mega 2560
  - For Hackerboard select ESP32 Arduino > DOIT ESP32 DEVKIT V1



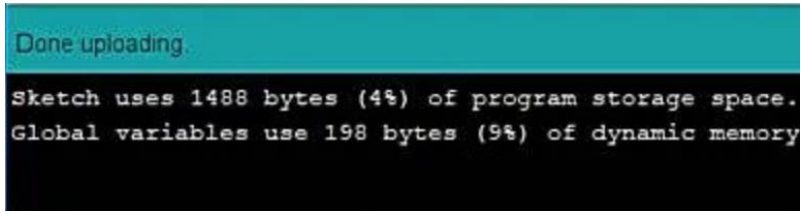


Activity

Uploading (Arduino IDE)

- Upload the code using the arrow on the top left corner of the IDE.

- The following message should appear o the IDE



Troubleshoot (Arduino IDE)

- For troubleshoot using the Arduino IDE, follow the steps in the presentation

MCR2_ArduinoIDE_Configuration_Windows_Ubuntu.pdf