



DesignNews

Arduino Pro Primer

Day 1:

Learning to Drive the Arduino Giga R1 WiFi

Sponsored by

DigiKey



Webinar Logistics

- Turn on your system sound to hear the streaming presentation.
- If you have technical problems, click “Help” or submit a question asking for assistance.
- Participate in ‘Attendee Chat’ by maximizing the chat widget in your dock.

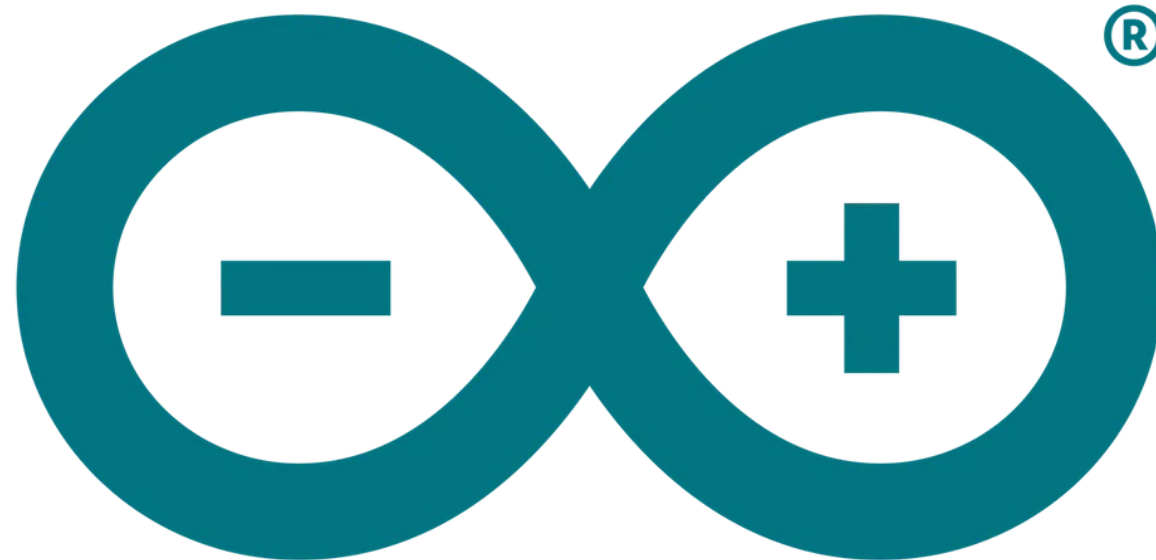


Fred Eady

Visit 'Lecturer Profile' in your console for more details.

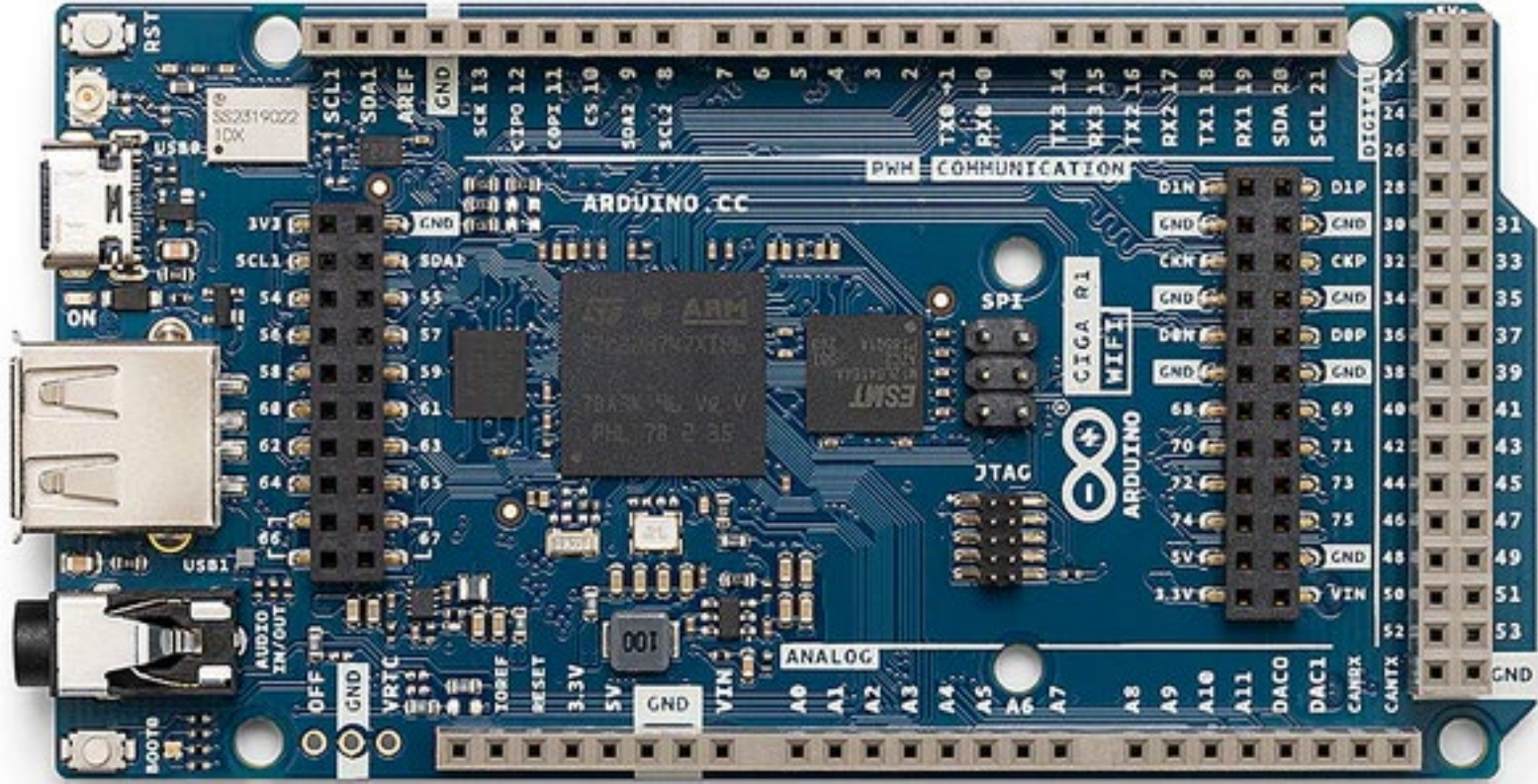
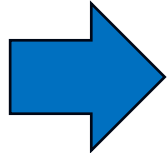
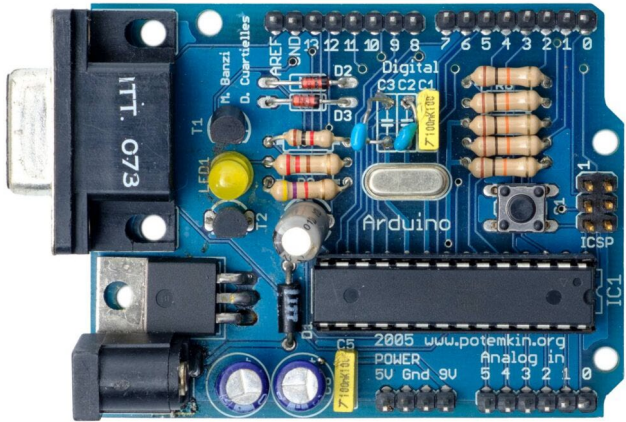
AGENDA

- **Arduino Development Using Visual Studio Code**
 - **Install Visual Studio Code and the Arduino Extension**
 - **GPIO/Blinky Sketch**
 - **Dual Core Sketches**

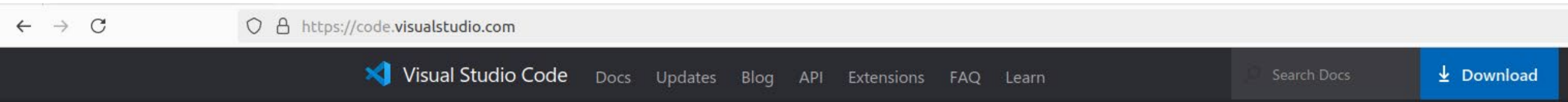


ARDUINO

From the Beginning



Install Visual Studio Code



[Version 1.90](#) is now available! Read about the new features and fixes from May.

Code editing. Redefined.

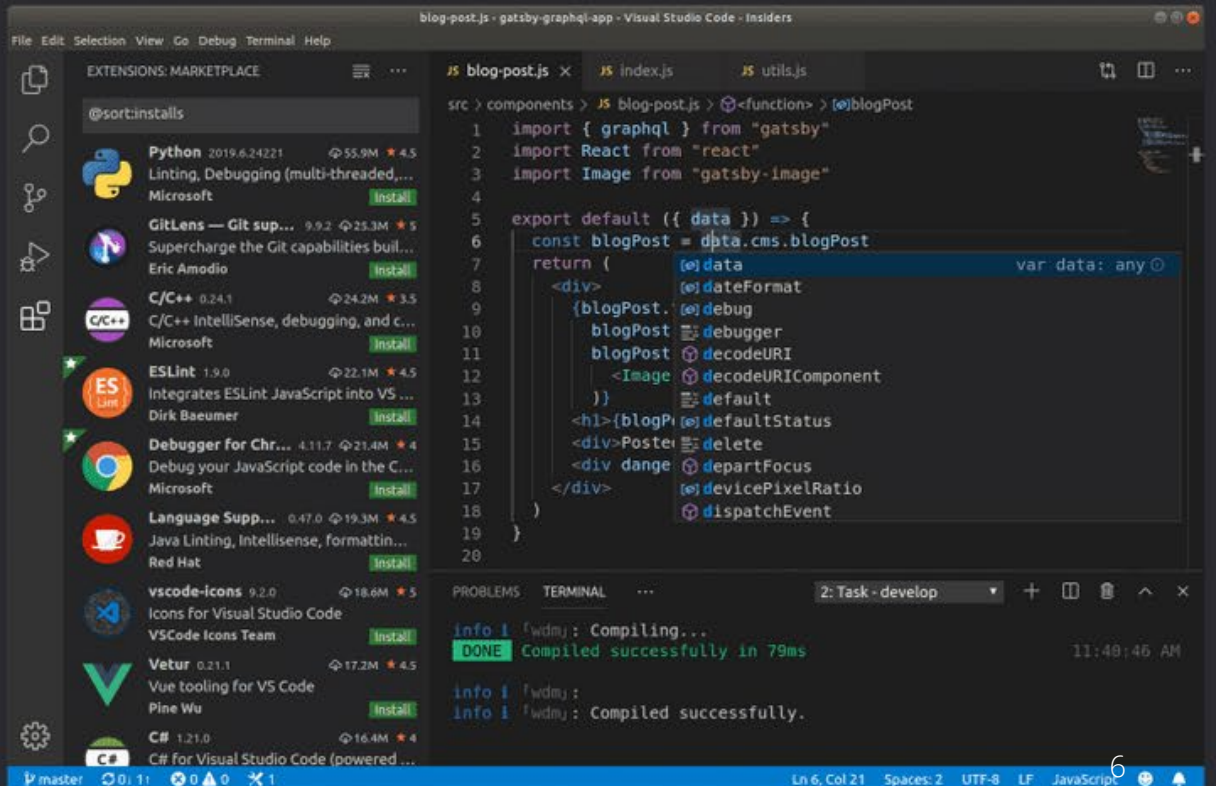
Free. Built on open source. Runs everywhere.

Download buttons for different operating systems:

- [.deb](#) (Debian, Ubuntu...)
- [.rpm](#) (Red Hat, Fedora...)

[Web](#), [Insiders edition](#), or [other platforms](#)

By using VS Code, you agree to its [license and privacy statement](#).



Install the Arduino Extension

Extension: Arduino - Visual Studio Code

File Edit Selection View Go Run Terminal Help

EXTENSIONS Search Extensions in Marketplace

INSTALLED 13

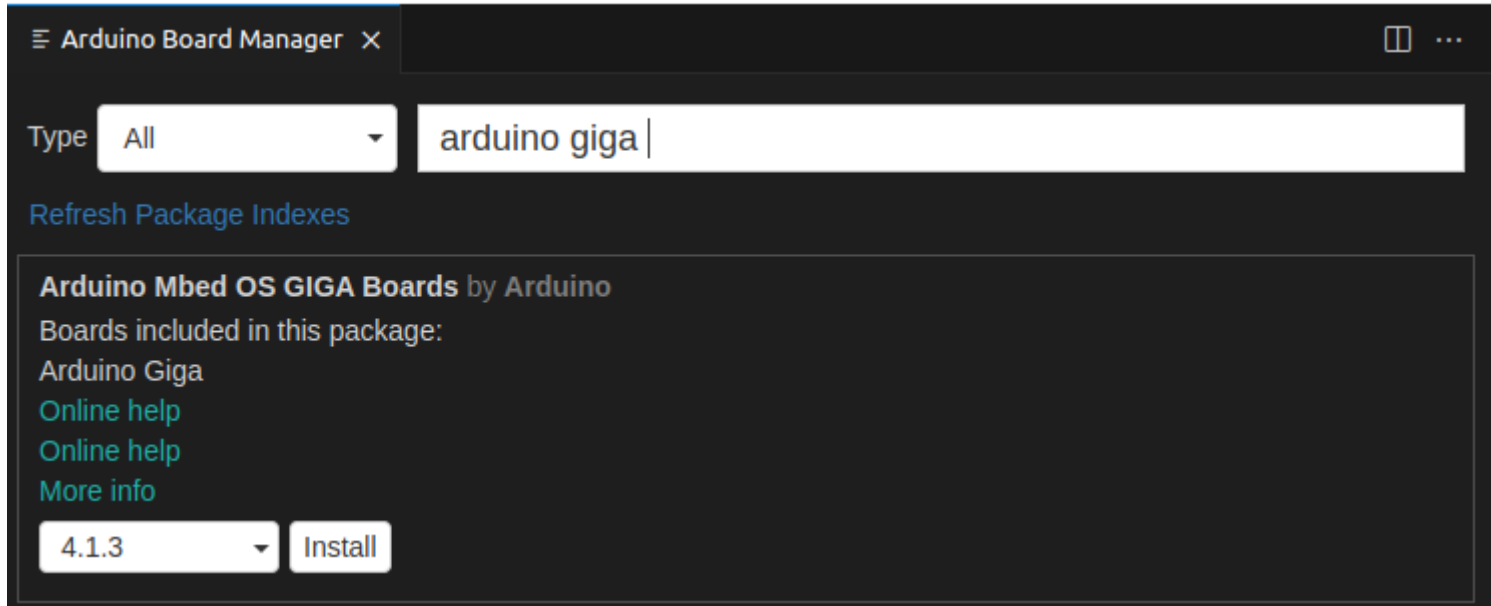
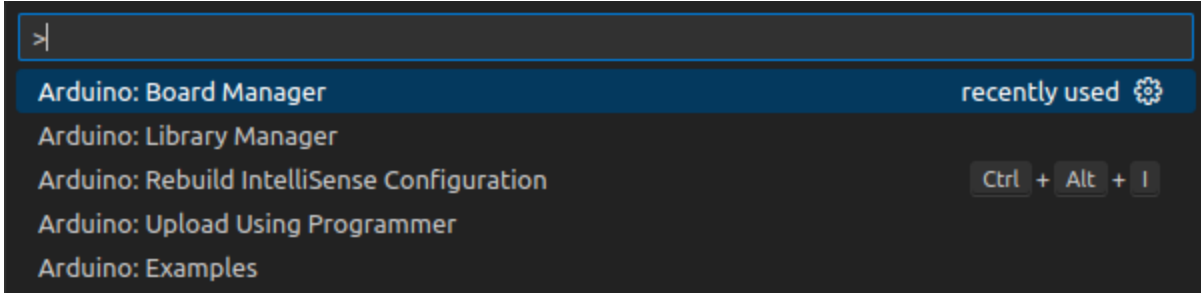
- Arduino** 2.1M ★ 3
Arduino for Visual Studio Code
Microsoft Install
- C/C++**
C/C++ IntelliSense, debugging, and ...
Microsoft
- CMake**
CMake language support for Visual S...
twxs

Arduino v0.6.0 Preview
Microsoft microsoft.com | 2,160,710 | ★★★★★ (125)
Arduino for Visual Studio Code
Install

DETAILS FEATURES CHANGELOG DEPENDENCIES

Visual Studio Code extension for Arduino

Install the Arduino Giga R1 WiFi Board Package



GPIO/Blinky Sketch

```
skeleton_sketch.ino - skeleton_sketch - Visual Studio Code  
File Edit Selection View Go Run Terminal Help  
EXPLORER  
SKELETON_SKETCH  
skeleton_sketch.ino 8  
skeleton_sketch.ino > setup()  
1 void setup() {  
2     // put your setup code here, to run once:  
3     pinMode(LED_B, OUTPUT);  
4 }  
5  
6 void loop() {  
7     // put your main code here, to run repeatedly:  
8     digitalWrite(LED_B, HIGH);  
9     delay(500);  
10    digitalWrite(LED_B, LOW);  
11    delay(500);  
12 }
```

Ln 1, Col 1 Spaces: 2 UTF-8 LF { } C++ <Select Programmer> <Select Board Type> <Select Serial Port> Linux

GPIO/Blinky Sketch – Select Arduino Giga R1 WiFi

skeleton_sketch.ino - skeleton_sketch - Visual Studio Code

File Edit Selection View Go Run Terminal Help

EXPLORER

SKELETON_SKETCH

skeleton_sketch.ino 8

```
1 void setup() {  
2     // put your setup code here, to run once:  
3     pinMode(LED_B, OUTPUT);  
4 }  
5  
6 void loop() {
```

Arduino Board Configuration

Selected Board:

- Select your board
- Adafruit Circuit Playground (Arduino AVR Boards)
- Arduino Yún Mini (Arduino AVR Boards)
- Arduino Industrial 101 (Arduino AVR Boards)
- Linino One (Arduino AVR Boards)
- Arduino Uno WiFi (Arduino AVR Boards)
- Arduino Giga R1 (Arduino Mbed OS GIGA Boards)

Ln 1, Col 1 Spaces: 2 UTF-8 LF {} C++ <Select Programmer> <Select Board Type> <Select Serial Port> Linux

GPIO/Blinky Sketch – Select Programmer

skeleton_sketch.ino - skeleton_sketch - Visual Studio Code

File Edit Selection View Go Run Terminal Help

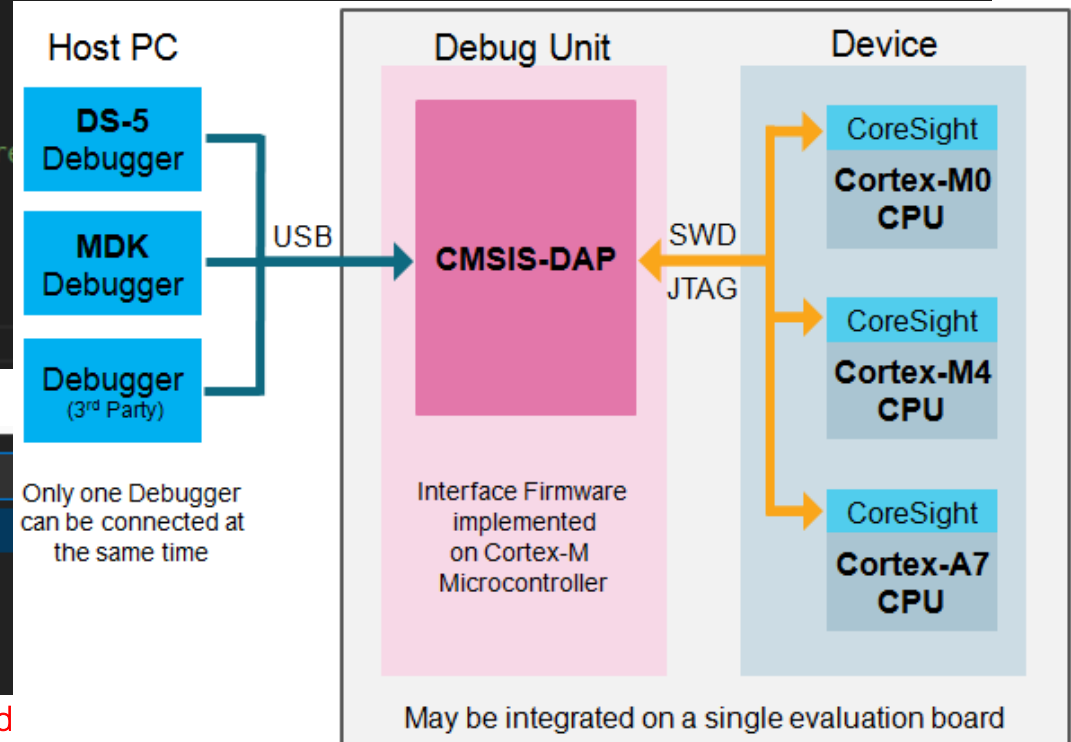
EXPLORER

- SKELETON_SKETCH
 - .vscode
 - skeleton_sketch.ino**

```

1  void setup() {
2      // put your setup code here, to run once:
3      pinMode(LED_B, OUTPUT);
4  }
5
6  void loop() {
7      // put your main code here
8      digitalWrite(LED_B, HIGH);
9      delay(500);
10     digitalWrite(LED_B, LOW);
11     delay(500);
12 }
    
```

CoreSight Debug Access Port (DAP)



Select programmer

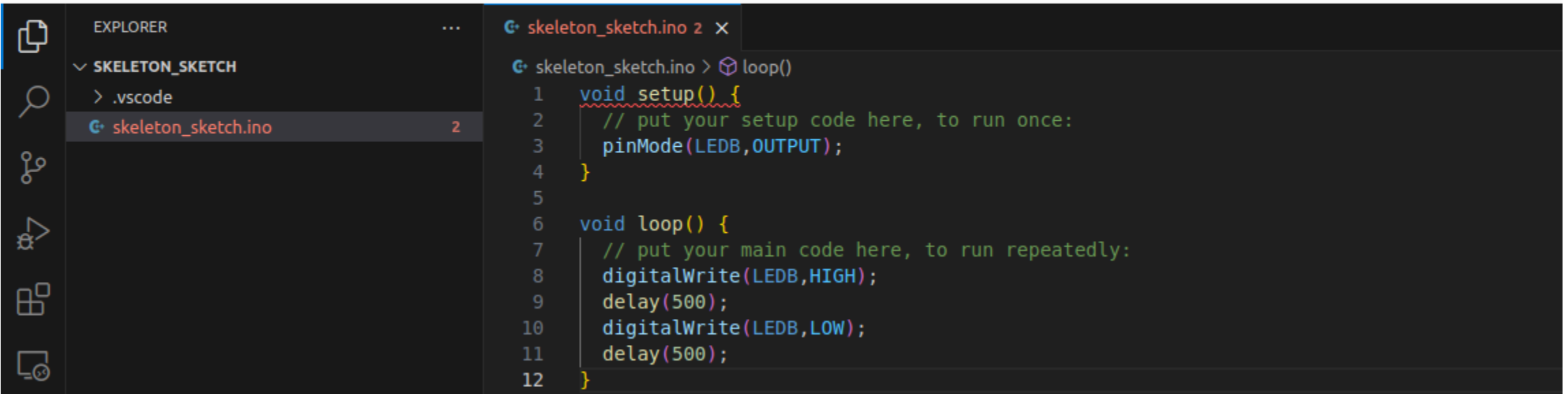
- ARM CMSIS-DAP compatible cmsis-dap**
- STMicroelectronics STLINK stlink
- JLink jlink
- BlackMagic blackmagic

Common Microcontroller Software Interface Standard

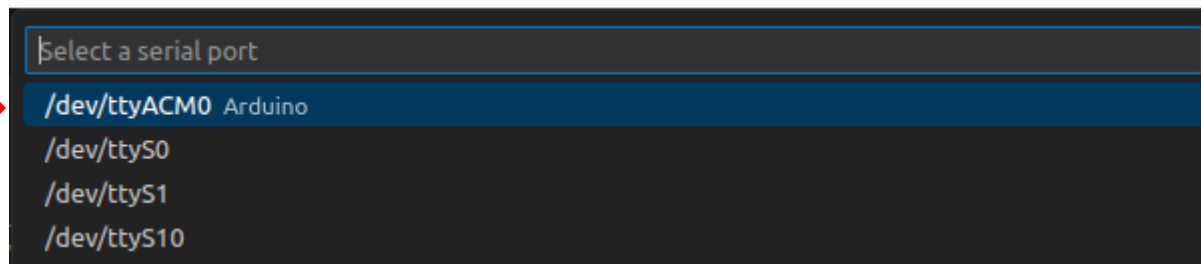
GPIO/Blinky Sketch – Select Serial Port

skeleton_sketch.ino - skeleton_sketch - Visual Studio Code

File Edit Selection View Go Run Terminal Help



```
1 void setup() {  
2     // put your setup code here, to run once:  
3     pinMode(LED_B, OUTPUT);  
4 }  
5  
6 void loop() {  
7     // put your main code here, to run repeatedly:  
8     digitalWrite(LED_B, HIGH);  
9     delay(500);  
10    digitalWrite(LED_B, LOW);  
11    delay(500);  
12 }
```



GPIO/Blinky Sketch – Verify the Sketch

skeleton_sketch.ino - skeleton_sketch - Visual Studio Code

File Edit Selection View Go Run Terminal Help

EXPLORER

SKELETON_SKETCH

.vscode

skeleton_sketch.ino 2

```
1 void setup() {
2   // put your setup code here, to run once:
3   pinMode(LED_B, OUTPUT);
4 }
5
6 void loop() {
7   // put your main code here, to run repeatedly:
8 }
```

arduino.json × skeleton_sketch.ino 2

```
.vscode > {} arduino.json > configuration
1 {
2   "configuration": "target_core=cm7,split=100_0",
3   "board": "arduino:mbed_giga:giga",
4   "programmer": "cmsis-dap",
5   "port": "/dev/ttyACM0",
6   "output": "/home/fred/skeleton_sketch/build",
7   "sketch": "skeleton_sketch.ino"
8 }
```

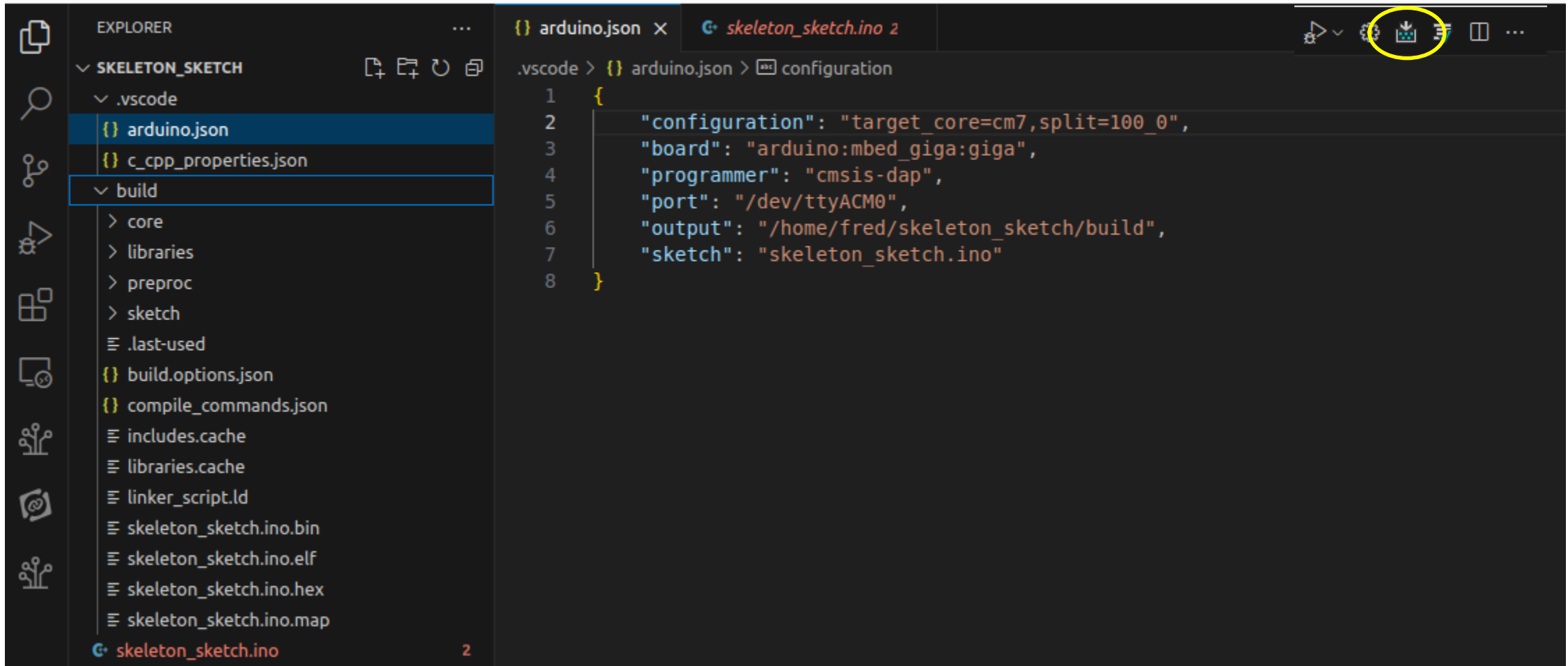
Ln 1, Col 1 Spaces: 2 UTF-8 LF {} C++ ARM CMSIS-DAP compatible Arduino Giga R1 /dev/ttyACM0

Added Manually

GPIO/Blinky Sketch – Download the Sketch

arduino.json - skeleton_sketch - Visual Studio Code

File Edit Selection View Go Run Terminal Help



The screenshot shows the Visual Studio Code interface. The Explorer view on the left shows the project structure for 'skeleton_sketch'. The file 'arduino.json' is selected. The editor view shows the content of 'arduino.json', which is a JSON configuration file for the Arduino IDE. The configuration includes the target core, board, programmer, port, output directory, and sketch file name.

```
.vscode > {} arduino.json > configuration
1  {
2    "configuration": "target_core=cm7,split=100_0",
3    "board": "arduino:mbed_giga:giga",
4    "programmer": "cmsis-dap",
5    "port": "/dev/ttyACM0",
6    "output": "/home/fred/skeleton_sketch/build",
7    "sketch": "skeleton_sketch.ino"
8  }
```

GPIO/Blinky Sketch – arduino.json - Board Configuration

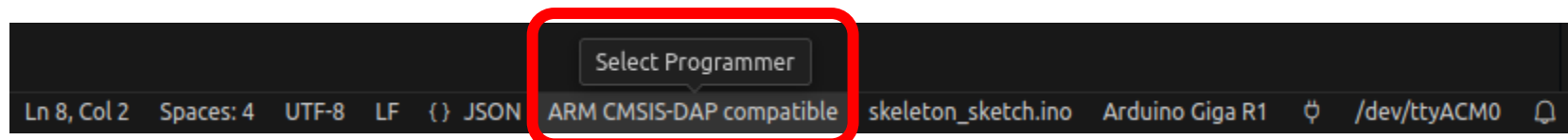
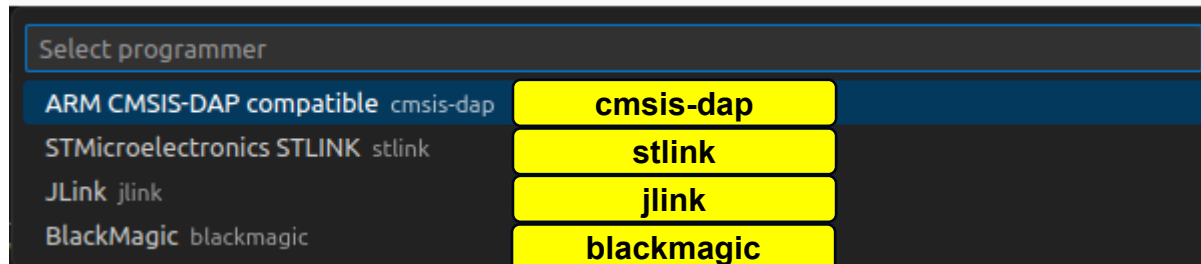
The image shows two screenshots of Visual Studio Code. The top screenshot shows the `arduino.json` file with a yellow arrow pointing to the `"configuration": "target_core=cm7,split=100_0"` line. The right-hand side shows the 'Arduino Board Configuration' window with 'Selected Board' set to 'Arduino Giga R1 (Arduino Mbed OS GIGA Boards)', 'Target core' set to 'Main Core', and 'Flash split' set to 'Main Core'. Yellow callouts link 'target_core=cm7' to the 'Main Core' option and 'target_core=cm4' to the 'M4 Co-processor' option.

The bottom screenshot shows the same `arduino.json` file with a yellow arrow pointing to the `"configuration": "target_core=cm7,split=100_0"` line. The right-hand side shows the 'Arduino Board Configuration' window with 'Selected Board' set to 'Arduino Giga R1 (Arduino Mbed OS GIGA Boards)', 'Target core' set to 'Main Core', and 'Flash split' set to '2MB M7 + M4 in SDRAM'. Yellow callouts link 'split=100_0' to the selected '2MB M7 + M4 in SDRAM' option, 'split=75_25' to the '1.5MB M7 + 0.5MB M4' option, and 'split=50_50' to the '1MB M7 + 1MB M4' option.

The screenshot shows the status bar at the bottom of VS Code. It displays 'Ln 8, Col 2', 'Spaces: 4', 'UTF-8', 'LF', '{ } JSON', 'ARM CMSIS-DAP compatible', 'skeleton_sketch.ino', 'Arduino Giga R1', and '/dev/ttyACM0'. A red circle highlights the 'Show Board Config' button next to the board name.

GPIO/Blinky Sketch – **arduino.json** - Select Programmer

```
{ } arduino.json x skeleton_sketch.ino
.vscode > { } arduino.json > configuration
1 {
2   "configuration": "target_core=cm7,split=50_50",
3   "board": "arduino:mbed_giga:giga",
4   "programmer": "cmsis-dap",
5   "port": "/dev/ttyACM0",
6   "output": "/home/fred/skeleton_sketch/build",
7   "sketch": "skeleton_sketch.ino"
8 }
```



Dual Core Sketches – **arduino.json** - M4 and M7

vscode > {} arduino.json > ...

```
1 {  
2   "configuration": "target_core=cm4,split=50_50",  
3   "board": "arduino:mbed_giga:giga",  
4   "programmer": "cmsis-dap",  
5   "port": "/dev/ttyACM0",  
6   "output": "/home/fred/rpc_sketch_M4/build"  
7 }
```

Arduino Board Configuration

Selected Board: Arduino Giga R1 (Arduino Mbed OS GIGA Boards)

Target core: M4 Co-processor

Flash split: 1MB M7 + 1MB M4



vscode > {} arduino.json > ...

```
1 {  
2   "configuration": "target_core=cm7,split=50_50",  
3   "board": "arduino:mbed_giga:giga",  
4   "programmer": "cmsis-dap",  
5   "port": "/dev/ttyACM0",  
6   "output": "/home/fred/rpc_sketch_M7/build"  
7 }
```

Arduino Board Configuration

Selected Board: Arduino Giga R1 (Arduino Mbed OS GIGA Boards)

Target core: Main Core

Flash split: 1MB M7 + 1MB M4

Dual Core Sketches – RPC (Remote Procedure Call)

```

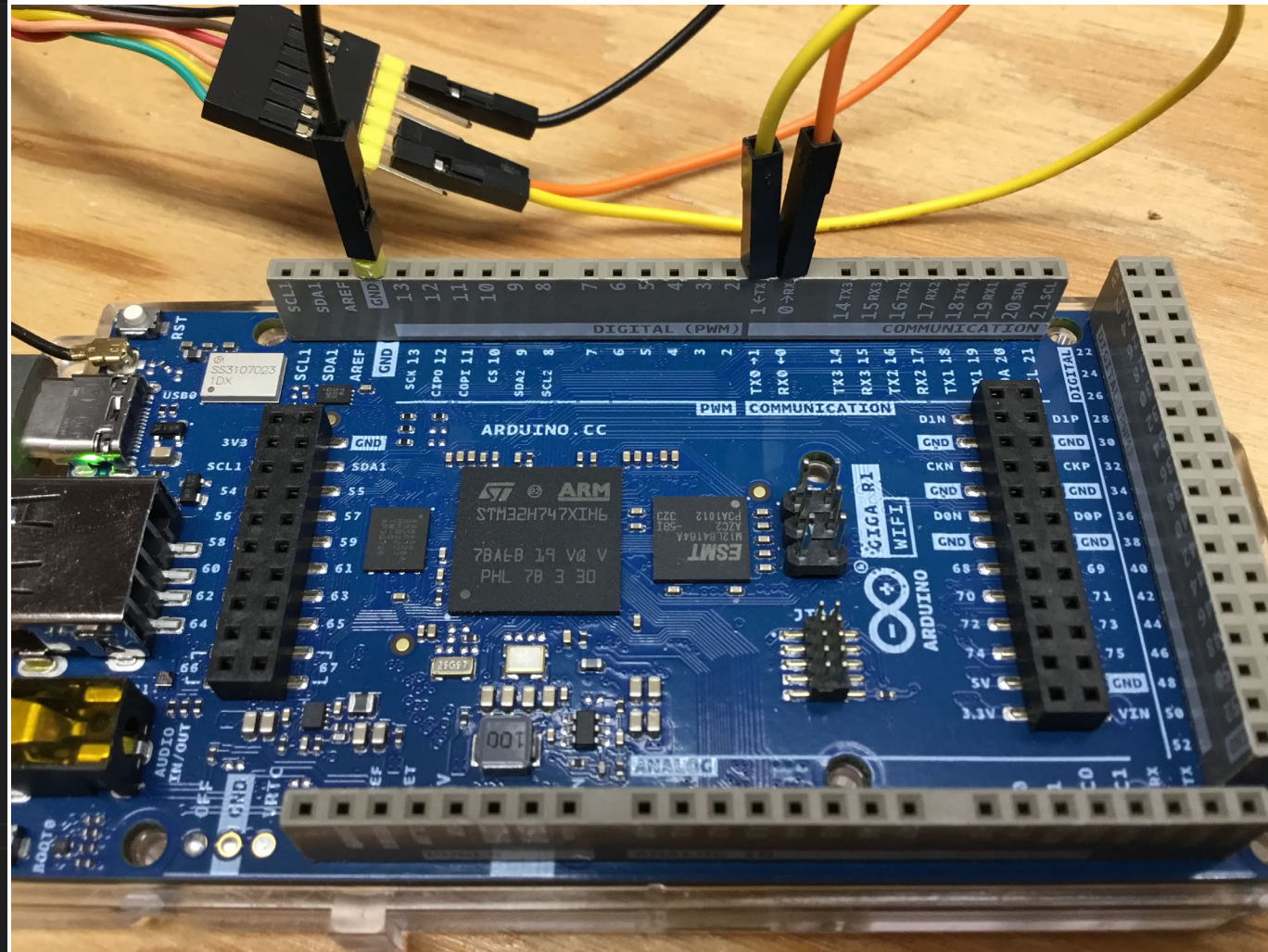
C: rpc_sketch_M7.ino 3 X
C: rpc_sketch_M7.ino > setup()
1 #include "Arduino.h"
2 #include "RPC.h"
3
4 String cmd;
5 String rpcBuf;
6
7 void setup() {
8   RPC.begin();
9   Serial1.begin(115200);
10 }
11
12 void loop() {
13   if(Serial1.available())
14   {
15     // get RGB command from Serial1
16     // RGB command must end with \n
17     cmd = Serial1.readStringUntil('\n');
18     if(cmd.equals("R"))
19     {
20       RPC.call("led_M4",'R');
21     }
22     if(cmd.equals("G"))
23     {
24       RPC.call("led_M4",'G');
25     }
26     if(cmd.equals("B"))
27     {
28       RPC.call("led_M4",'B');
29     }
30   }
31   // get and print command feedback from M4
32   while(RPC.available())
33   {
34     rpcBuf += (char)RPC.read();
35   }
36   if(rpcBuf.length() > 0)
37   {
38     Serial1.print(rpcBuf);
39     rpcBuf = "";
40   }
41 }

```

```

C: rpc_sketch_M4.ino 3 X
C: rpc_sketch_M4.ino > setup()
1
2 #include "Arduino.h"
3 #include "RPC.h"
4
5 void led_M4(char cmdChar)
6 {
7   if(cmdChar == 'R')
8   {
9     digitalWrite(LED_R,LOW);
10    digitalWrite(LED_G,HIGH);
11    digitalWrite(LED_B,HIGH);
12    RPC.println("RED LED IS ON");
13  }
14  if(cmdChar == 'G')
15  {
16    digitalWrite(LED_R,HIGH);
17    digitalWrite(LED_G,LOW);
18    digitalWrite(LED_B,HIGH);
19    RPC.println("GREEN LED IS ON");
20  }
21  if(cmdChar == 'B')
22  {
23    digitalWrite(LED_R,HIGH);
24    digitalWrite(LED_G,HIGH);
25    digitalWrite(LED_B,LOW);
26    RPC.println("BLUE LED IS ON");
27  }
28 }
29
30 void setup() {
31   pinMode(LED_R,OUTPUT);
32   pinMode(LED_G,OUTPUT);
33   pinMode(LED_B,OUTPUT);
34   RPC.bind("led_M4",led_M4);
35   RPC.begin();
36 }
37
38 void loop() {
39 }

```



Dual Core Sketches – RPC (Remote Procedure Call)

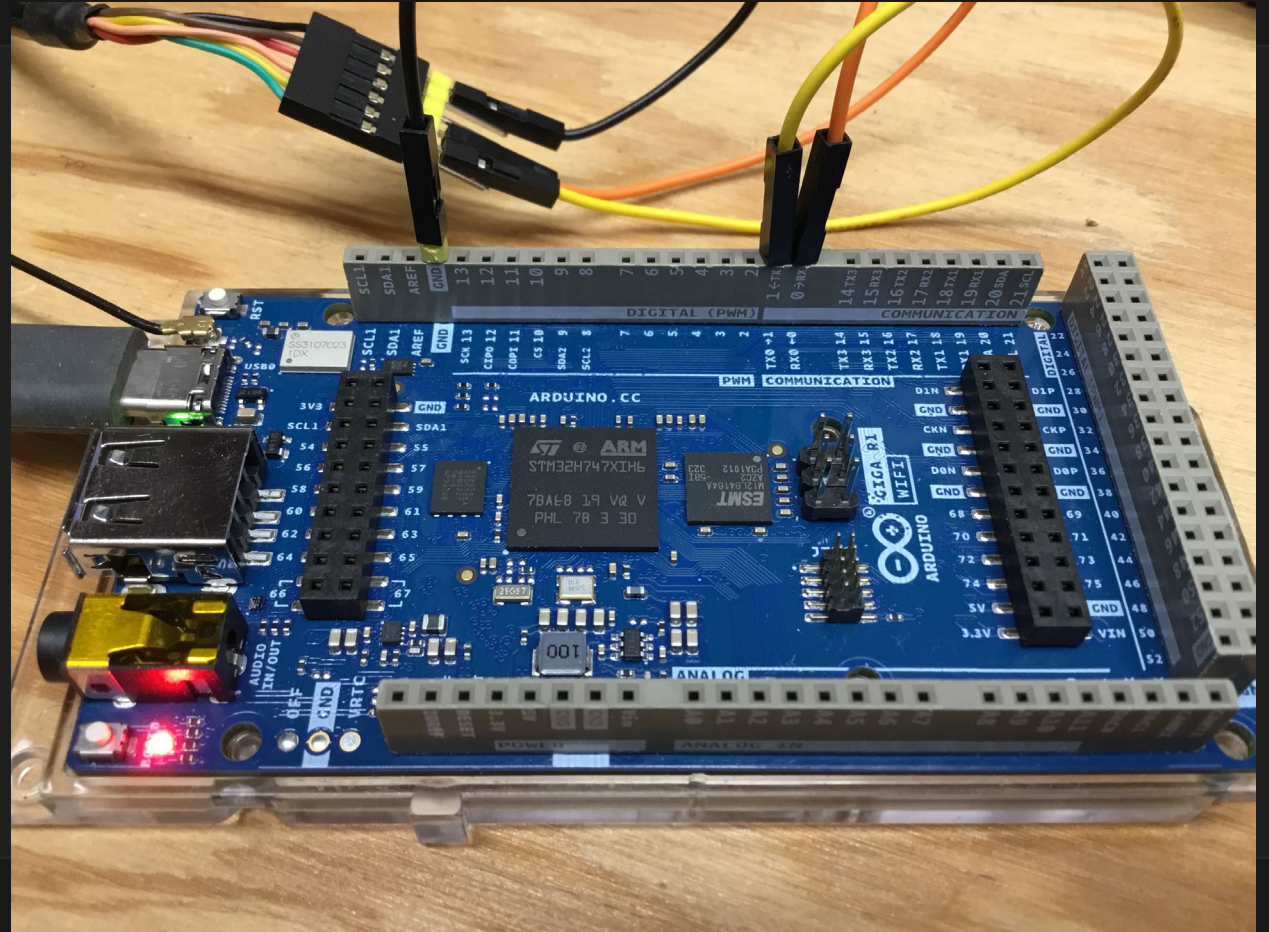
```
rpc_sketch_M4.ino 3 ●
rpc_sketch_M4.ino > setup()
1
2 #include "Arduino.h"
3 #include "RPC.h"
4
5 void led_M4(char cmdChar)
6 {
7   if(cmdChar == 'R')
8   {
9     digitalWrite(LED_R, LOW);
10    digitalWrite(LED_G, HIGH);
11    digitalWrite(LED_B, HIGH);
12    RPC.println("RED LED IS ON");
13  }
14  if(cmdChar == 'G')
15  {
16    digitalWrite(LED_R, HIGH);
17    digitalWrite(LED_G, LOW);
18    digitalWrite(LED_B, HIGH);
19    RPC.println("GREEN LED IS ON");
20  }
21  if(cmdChar == 'B')
22  {
```

PROBLEMS 3 OUTPUT DEBUG CONSOLE TERMINAL PORTS SERIAL MONITOR

+ Open an additional monitor

Monitor Mode Serial View Mode Text Port /dev/ttyUSB0 - FTDI Baud rate 115200 Line ending None Stop Monitoring

```
---- Opened the serial port /dev/ttyUSB0 ----
---- Sent hex encoded message: "520A" ----
RED LED IS ON
```



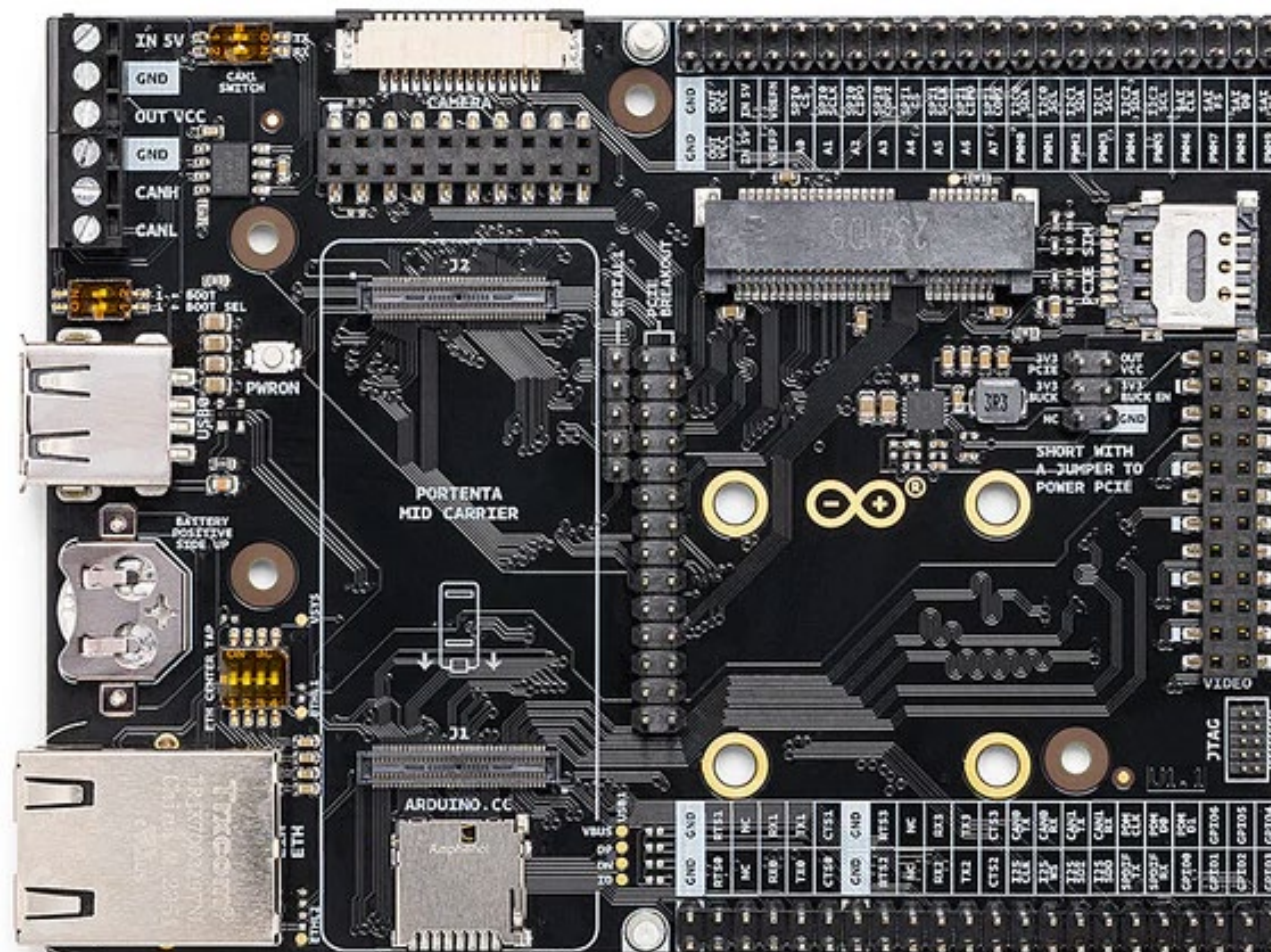
Next Time...

MORE TO COME..

Thank you for attending!!!

Please consider the resources below:

- [Today's Download Package](#)
- [arduino.cc](https://www.arduino.cc)





Thank You

Sponsored by

