



COMMU Module Extend RS485/TTL CAN/I2C Port

SKU: M011

COMMU is a Multi-Communication-Interface-Converter. Integrated with *2IIC*, *1TTL*, *1CAN*, *1RS485*. Apparently **COMMU** has packed with most of series communications. Default connection: TTL - UART0, RS485 - UART2. Since ESP32 pin map is allowed for re-assign, you can re-assign or re-mapping the TTL or RS485 interface to other pins.

Product Features

- 2x I2C Interface
- 1x CAN Interface
- 1x RS485 Interface
- 1x TTL Interface
- CAN controller: MCP2515-1/SO
- RS485 Transceiver: SP3485EN-L/TR

Kit includes

- 1x M5Stack **COMMU** Module

Example

Arduino IDE
CAN communication

These are two COMMU examples for CAN communication, transmitter and receiver. Press Button A to sent the message, and display the received message on the screen.

Step 1: Copy [MCP_CAN lib file](#) to C:\Users\<<user_name>\Documents\Arduino\libraries, **Step 2:** Open project file `commu_can_transmitter.ino`, and `commu_can_receiver.ino` **Step 3:** Compile and upload the two project to two M5Cores separately.

Below code is incomplete. If you want the complete code, please click [here](#).

```
/*
   commu_can_transmitter.ino
*/
#include <M5Stack.h>
#include <mcp_can.h>
#include "m5_logo.h"

#define CAN0_INT 15 // Set INT to pin 2
MCP_CAN CAN0(12); // Set CS to pin 10

// declaration
byte data[8] = {0x00, 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07};

// initialization
M5.begin();
CAN0.begin(MCP_ANY, CAN_1000KBPS, MCP_8MHZ);
/* Change to normal mode to allow messages to be transmitted */
CAN0.setMode(MCP_NORMAL);

// send data
CAN0.sendMsgBuf(0x100, 0, 8, data);
/*
   commu_can_receiver.ino
*/
#include <M5Stack.h>
#include <mcp_can.h>
#include "m5_logo.h"

#define CAN0_INT 15 // Set INT to pin 2
MCP_CAN CAN0(12); // Set CS to pin 10

// declaration
byte data[8] = {0x00, 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07};

// initialization
M5.begin();
/* Initialize MCP2515 running at 16MHz with a baudrate of 500kb/s */
/* and the masks and filters disabled. */
CAN0.begin(MCP_ANY, CAN_1000KBPS, MCP_8MHZ);
/* Set operation mode to normal so theMCP2515 sends acks to received data. */
```

```
CAN0.setMode(MCP_NORMAL);  
pinMode(CAN0_INT, INPUT);// Configuring pin for /INT input  
  
// read data  
CAN0.readMsgBuf(&rxId, &len, rxBuf);
```

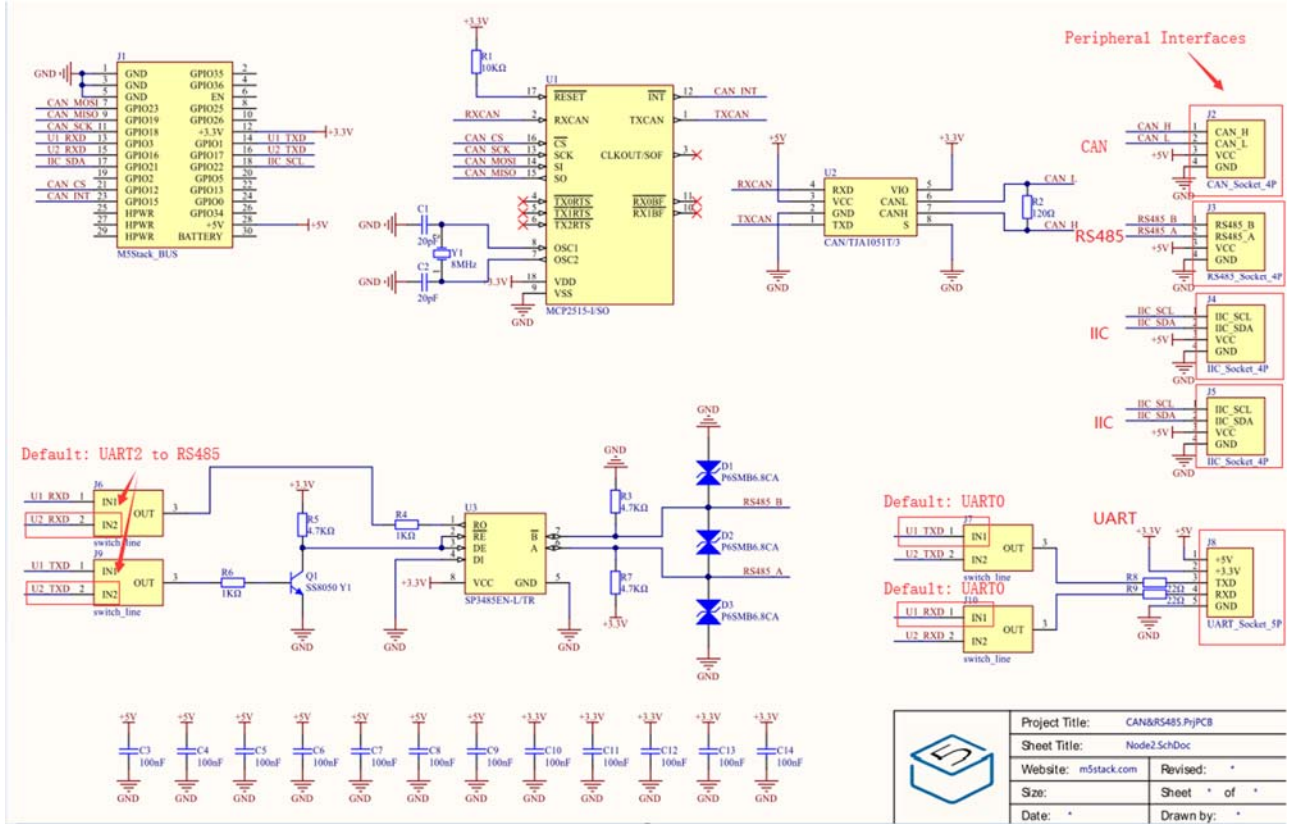


PinMap

<i>CAN</i>	<i>ESP32 Chip</i>
CAN_CS	GPIO12
CAN_INT	GPIO15
CAN_SCK	GPIO18

CAN	ESP32 Chip
CAN_MISO	GPIO19
CAN_MOSI	GPIO23
I2C Interface	ESP32 Chip
IIC_SDA	GPIO21
IIC_SCL	GPIO22

Schematic



	Project Title: CAN&RS485 PjPCB
	Sheet Title: Node2.SchDoc
	Website: m5stack.com
	Size: Sheet ' of '
	Date: ' ' Drawn by: ' '

