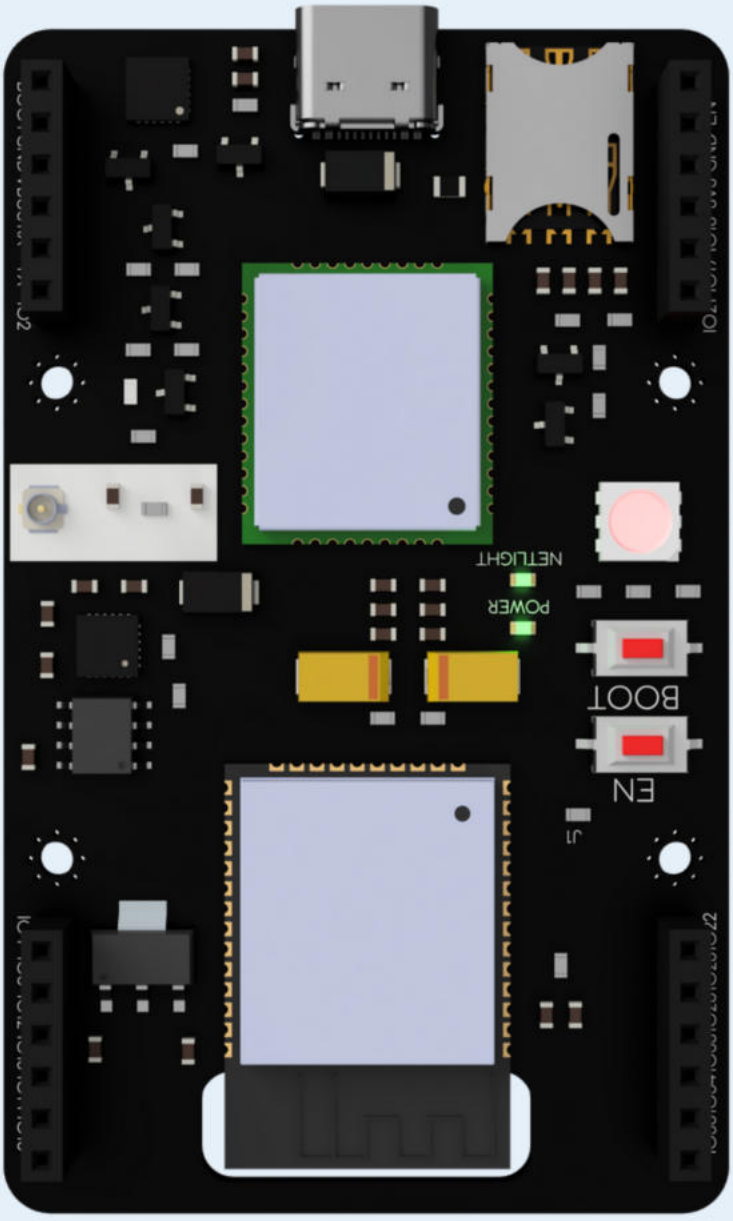
MICROMIS BASE V1

QUICK START GUIDE



Made by:
DEVICE
PROTOTYPE

WELCOME

The Micromis Base V1 developer board is a modern tool for engineers and programmers to create advanced electronic projects. The main feature of the board is the use of the ESP32 chip, which is one of the most popular chips for creating projects using wireless networks (Wi-Fi and Bluetooth). This makes the board ideal for creating Internet of Things (IoT) devices and other applications requiring a wireless connection. Using Micromis is facilitated by a built-in USB-UART converter, which allows the device to be programmed using a USB-C cable. A USB socket built into the device also allows powering the device's components and additional components connected to the platform.

The platform is equipped with a Quectel M65 modem, which enables connectivity to cellular networks and data transmission over GSM networks. The modem has an integrated antenna connector, so it can be easily connected to an external antenna for better connection quality.

The device also has an addressable LED, which can be software-controlled and used to visualize the device's status or to create lighting effects. In addition, it has been equipped with the MPU6050 chip, which can measure acceleration and rotation in three axes, allowing the creation of motion-sensing designs.

The board has also been equipped with the LM75 temperature sensor, which allows the measurement of ambient temperature with an accuracy of 0.5 degrees Celsius. This is useful for applications that require temperature measurement, such as air-conditioning systems and measuring devices.

The Micromis Base V1 also features female goldpin leads, which allow the connection of external peripherals and Micromis overlays to expand the capabilities of the board itself.

The platform is also equipped with a number of protections, including overvoltage, short-circuit, over-temperature and over-current protection from the USB port, making it a suitable tool for electronics beginners.

HAVE FUN WHILE USING THE MICROMIS BASE V1!

Made by:
DEVICE PROTOTYPE

MICROMIS BASE V1: QUICK START

Using the Micromis Base V1 platform is extremely easy! To get started with your board, you need to follow the few steps below:

1. Unpack your Micromis Base V1 board from the packaging
2. Insert an active nanoSIM card into the SIM card slot
3. Connect the GSM antenna to the U.FL connector
4. Connect one side of the USB Type C cable to the Micromis Base V1 board and the other to the computer
5. Install the environment on your computer in which you will program the board
6. Install drivers for CP2102 chip from www.silabs.com/developers/usb-to-uart-bridge-vcp-drivers
7. Install data packages for ESP32 chips
8. Select the "ESP32 Dev Module" board
9. Upload your first program to the Micromis Base V1 board

If you have previously used boards with an embedded ESP32 chip in your development environment, you probably won't need to do any additional configuration, and the Micromis Base V1 board will work as soon as you connect it to your computer.

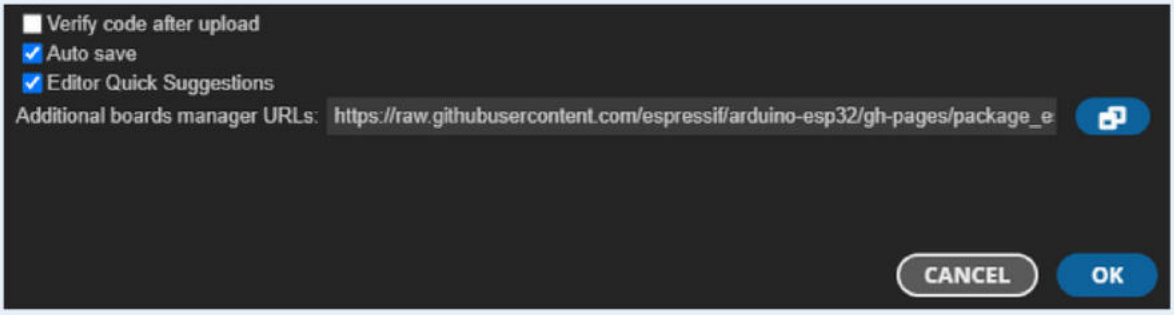
If you don't yet have a programming environment with which you will program the Micromis Base V1 board, or you don't know how to install data packages for boards with ESP32 chips, then on the following pages we will discuss the two most popular environments and how to get the Micromis Base V1 board operable with them.

MICROMIS BASE V1: USING WITH ARDUINO IDE

Arduino IDE is the most popular environment used mainly for hobby purposes. Due to the ability to import additional boards and the extremely large community of users of this IDE, many owners of boards with the ESP32 chip have decided to use this environment.

If you do not have the Arduino IDE environment installed then you need to download it from the link below and install it on your computer, preferably download version 2.0 or later.
https://www.arduino.cc/en/software

After installing the Arduino IDE environment, you need to click:
File -> Preferences and in the "Additional boards manager URLs" field enter the following link, this is a link to the official package from the manufacturer of the ESP32 chip:
https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json



After pasting the board manager link, you need to click on the "OK" button to exit the environment preferences. Now you need to click in turn:
Tools -> Board -> Boards Manager and in the board manager type "esp32" into the search engine, after a while you should see the package "esp32 by Espressif Systems", at the bottom of the box you need to click "Install", the latest version of ESP32 chip-equipped board packages will automatically install. If you don't see tile packages after adding the package link to the "Additional boards manager URLs" field and typing the phrase "esp32" in the tile manager search engine, it's a good idea to restart the whole environment.

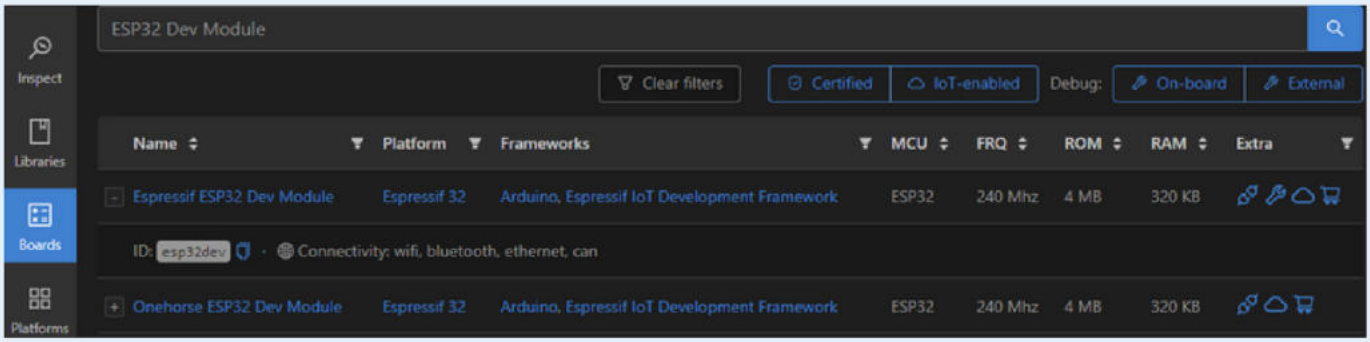
MICROMIS BASE V1: USING WITH VISUAL STUDIO CODE

The second most popular environment for programming boards equipped with ESP32 chips is Visual Studio Code with the PlatformIO IDE extension. The PlatformIO extension allows us to work comfortably with a huge number of development boards and standalone chips, which we can program in many frameworks. To use the capabilities of this environment, you must first download and install Visual Studio Code from the link: https://code.visualstudio.com/

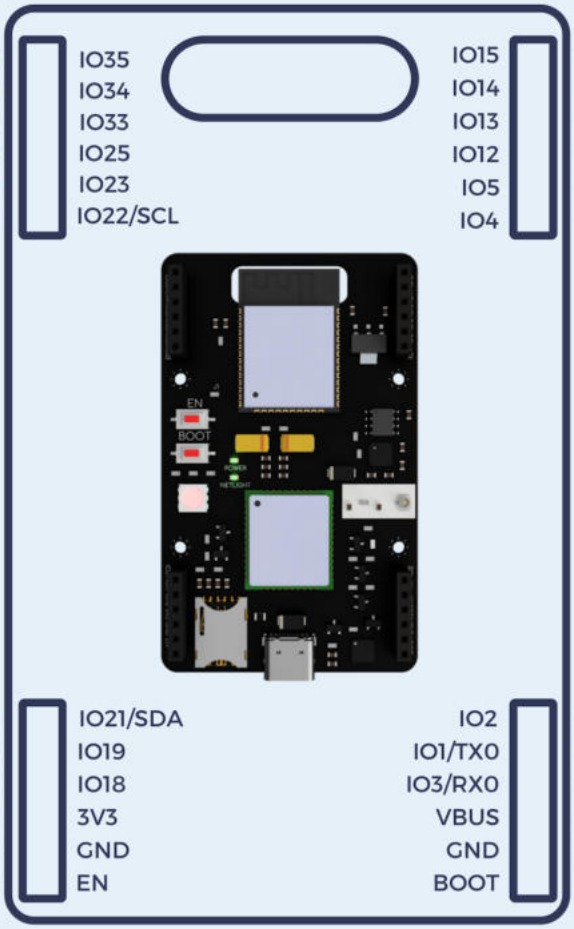
In addition, you should download and install Python 3.8.5 or later from the link: https://www.python.org/downloads/

Once you have installed the Visual Studio Code environment and Python, click on View -> Extension in Visual Studio Code, an extension browser window should open on the left. In the extension browser you need to type "PlatformIO IDE", when you click on the item with the name "PlatformIO IDE" a window will open with the details of the extension, now you just need to click "Install" and the extension will appear download and install itself.

After installing the extension, we need to click on the PlatformIO icon located on the toolbar on the left, and then click on the home icon on the bottom bar, which will bring up the extension's home page. Once you are in the extension's home page, you need to click on "Boards" and type "ESP32 Dev Module" in the tile search box. The board you are interested in will itself appear below the search box. When you create a project, all you have to do is copy the ID of the specific board and paste it into the project, or when generating the project, select the board you will program as "ESP32 Dev Module".



MICROMIS BASE V1: PIN FUNCTION



INPUT ONLY | RTC WAKEUP | ADC
INPUT ONLY | RTC WAKEUP | ADC
INPUT ONLY | RTC WAKEUP | ADC
INPUT ONLY | RTC WAKEUP | ADC
SPI MOSI
I2C SCL

ADC | RTC WAKEUP | TOUCH
ADC | RTC WAKEUP | TOUCH
ADC | RTC WAKEUP | TOUCH
ADC | RTC WAKEUP | TOUCH
SPI CS
ADC | RTC WAKEUP | TOUCH

I2C SDA
SPI MISO
SPI CLK
3V3
GND
EN

ADC | RTC WAKEUP | TOUCH
MAIN UART TX
MAIN UART RX
5V
GND
BOOT

ADC
Inputs for the ADC, the ADC has 12-bit resolution. With it, we can read analog values from 0 to 4095 in voltage ranges from 0V to 3.3V, where 0 is 0V and 4095 is 3.3V. Remember not to connect a voltage higher than 3.3V to the analog pins

RTC WAKEUP
The ESP32 chip supports waking up from an external source via an ultra-saving RTC chip using pins labeled RTC WAKEUP.

TOUCH
The ESP32 has built-in 10 internal capacitive touch sensors. They allow sensing the change in surfaces that have electrical charges. With this, we can create simple touch pads that can also be used to wake up the chip.

I2C
The ESP32 has two I2C channels and each pin can be set as SDA or SCL for ease of use, the components on the board and the leads on the goldpins have been routed to pins 21 (SDA) and 22 (SCL).

SPI
To communicate with external components we can use the SPI protocol built into the ESP32, on the board pins 23 (MOSI), 19 (MISO), 18 (CLK), 5 (CS) have been assigned to the SPI interface.

INPUT ONLY
The pins of the board marked INPUT ONLY do not allow us to control external components, we can use them to read analog or digital signals.

MAIN UART
The board's pins labeled MAIN UART allow communication via the UART protocol, are connected to the ESP32's main UART protocol, and can be used to program the chip bypassing the CP2102 chip built into the board. We do not recommend using these connectors for purposes other than UART communication.

3V3
3.3V power output, which can be used to power external components, but the current capacity of this connector is 350mA, if you need to power a more demanding component, use an external power source.

5V
5V power connector, which can be used to power external components, but the current capacity of this connector is 250mA, if you need to power a more demanding component, use an external power source. The connector can also be used to power the board in case the device is not powered from the USB port.

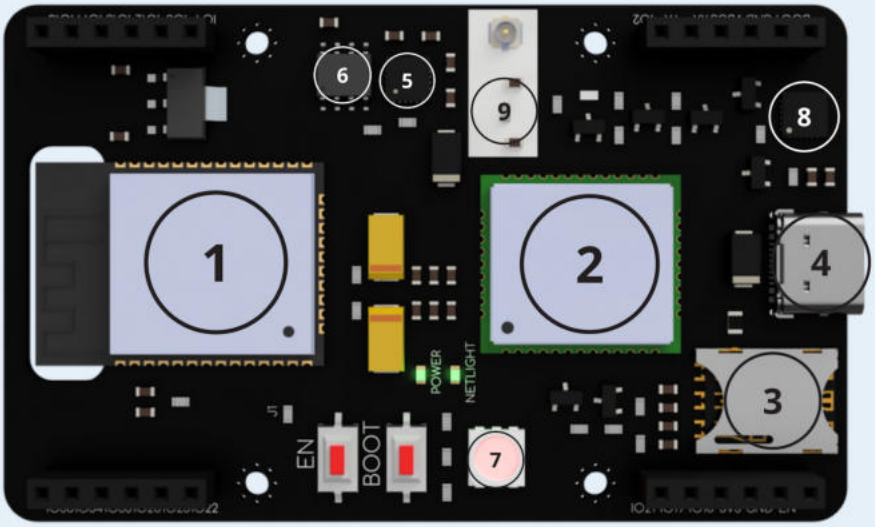
GND
Board pins for ground potential output.

BOOT
The BOOT pin is responsible for controlling the operating mode of the ESP32, thanks to it the chip can enter programming mode. The pin is connected to the BOOT button on the board.

EN
The EN pin is responsible for resetting the ESP32 chip. The pin is connected to the EN button on the board.

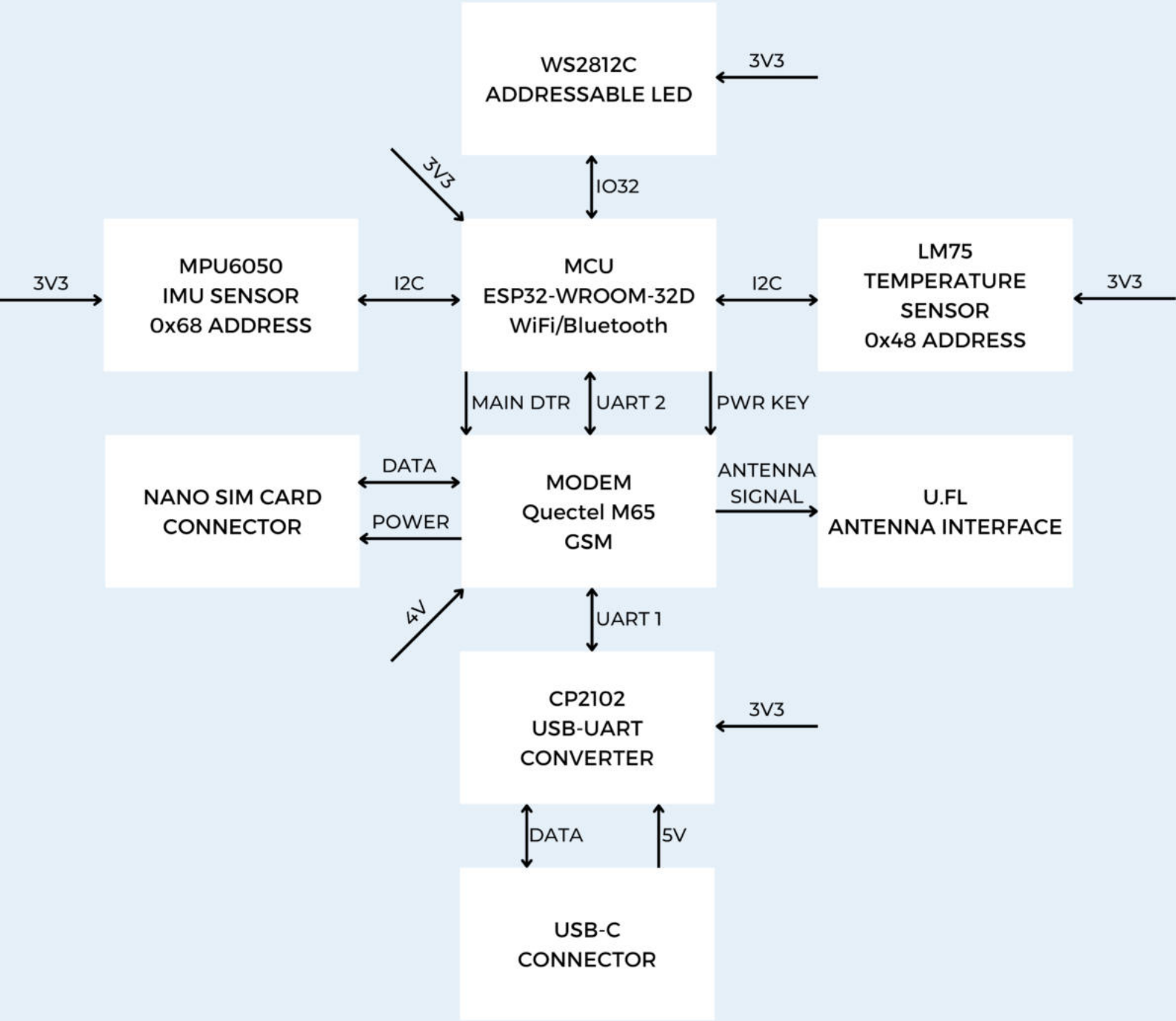
Made by:
DEVICE PROTOTYPE

MICROMIS BASE V1: IMPORTANT COMPONENTS ON BOARD

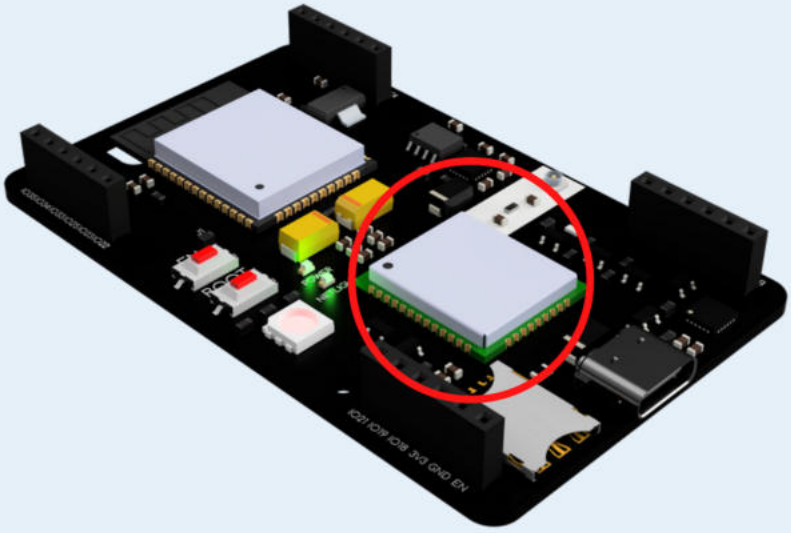


1. ESP32-WROOM-32D microcontroller
2. Quectel M65 GSM modem
3. NanoSim card slot
4. USB Type-C connector
5. MPU6050 accelerometer and gyroscope
6. LM75 temperature sensor
7. WS2812C addressable LED
8. CP2102 programming chip
9. Integrated GSM antenna array

MICROMIS BASE V1: BLOCK DIAGRAM OF KEY COMPONENTS



MICROMIS BASE V1: USING BUILT-IN COMPONENTS - GSM MODEM



The Micromis Base V1 development board has a built-in Quectel M65 modem for GSM network communication, which allows the device to connect to the Internet without WiFi and send SMS messages.

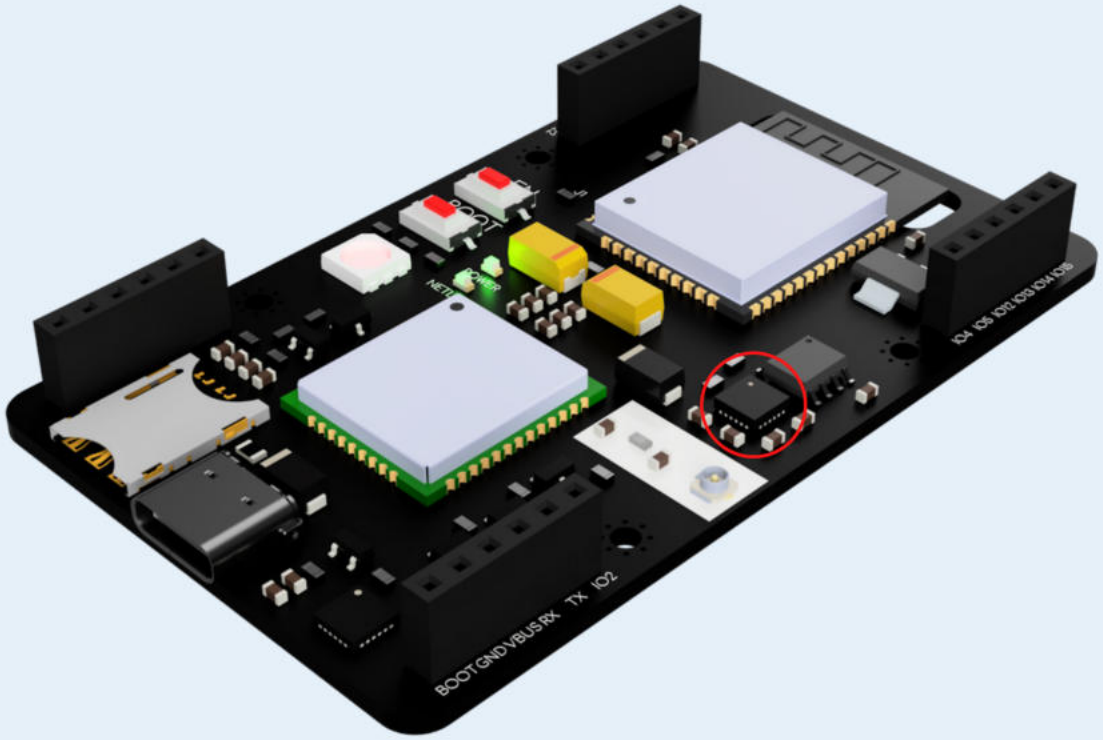
For proper operation of the modem we need an active nanoSIM size card and an antenna with a U.FL connector suitable for operation in the frequency band from 800MHz to 1900 MHz. Depending on our needs, we can use a SIM card that only allows mobile data exchange, there is no need for a SIM card with SMS and phone call support.

The UART protocol by which the modem communicates with the ESP32 is permanently connected to pins 16 (RX2 ESP32) and 17 (TX2 ESP32), which are the default port for the UART2 protocol on the ESP32 chip.

For easy management of the modem's operation, we can control the PWR_KEY and MAIN_DTR pins. The modem's PWR_KEY pin allows the modem to be turned on and off, when a high state is applied to ESP32 pin 27 for one second the modem will change its state from off to on or from on to off. When a high state is given for 20 ms on pin 26 of the ESP32, we activate the MAIN_DTR pin, which allows the modem to wake up when power saving is activated.

The board's built-in NETLIGHT LED indicates the modem's operation, if it blinks it means the modem is working, if not it means it is off.

MICROMIS BASE V1: USING BUILT-IN COMPONENTS - MPU6050 IMU

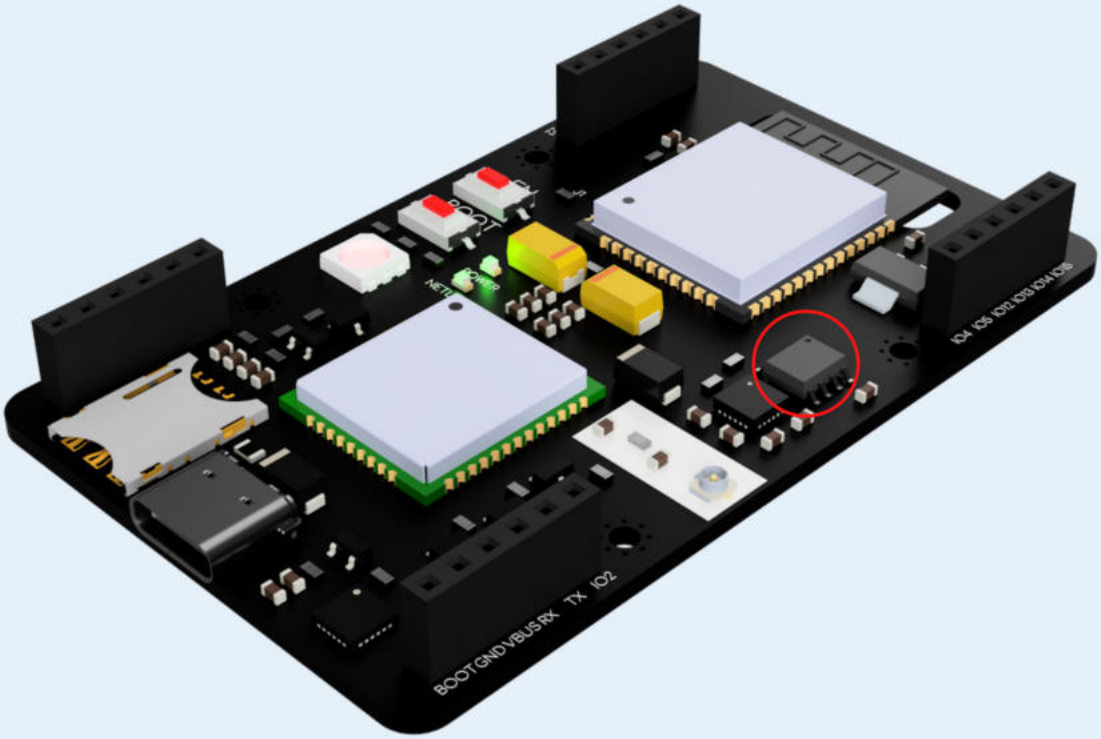


On the Micromis Base V1 development board is the MPU6050 chip, which can read acceleration and spatial orientation - a combination of a gyroscope and accelerometer.

The MPU6050 communicates with the ESP32 using the I2C protocol, which is also brought out on the Micromis device pins - pins 22 (SCL) and 21 (SDA). In order to communicate with the IMU, we will need its address - in the case of the chip embedded in the Micromis Base V1 board, the chip address cannot be changed - it is fixed at 0x68.

The chip allows for operation in different measurement ranges:
- accelerometer - ±2 g, ±4 g, ±8 g, ±16 g
- gyroscope - ±250 °/s, ±500 °/s, ±1000 °/s, ±2000 °/s

MICROMIS BASE V1: USING BUILT-IN COMPONENTS - LM75 TEMP SENSOR

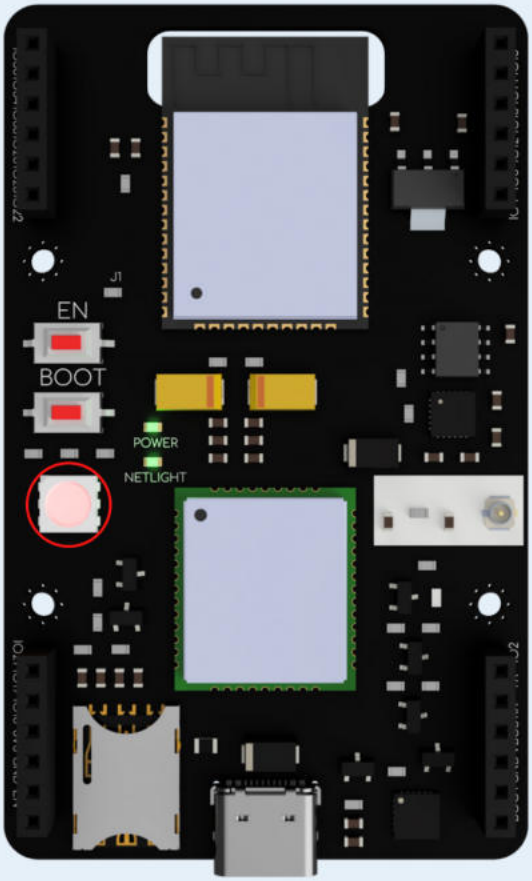


In addition to the MPU6050 chip, an LM75 temperature sensor is mounted on the Micromis Base V1 development board, which allows reading ambient temperatures from -55 ˚C to +125 ˚C.

The LM75 sensor communicates with the ESP32 using the I2C protocol, which is also brought out on the pins of the Micromis device - pins 22 (SCL) and 21 (SDA). In order to communicate with the LM75, we will need its address - in the case of the chip embedded in the Micromis Base V1 board, the address of the chip cannot be changed - it is fixed and is 0x48.

The LM75 temperature sensor allows us to control its state so that the sensor can be turned off at any time. A very important advantage is its low standard current consumption during operation (250µA) and while it is programmed off (4µA).

MICROMIS BASE V1: USING BUILT-IN COMPONENTS - WS2812C LED

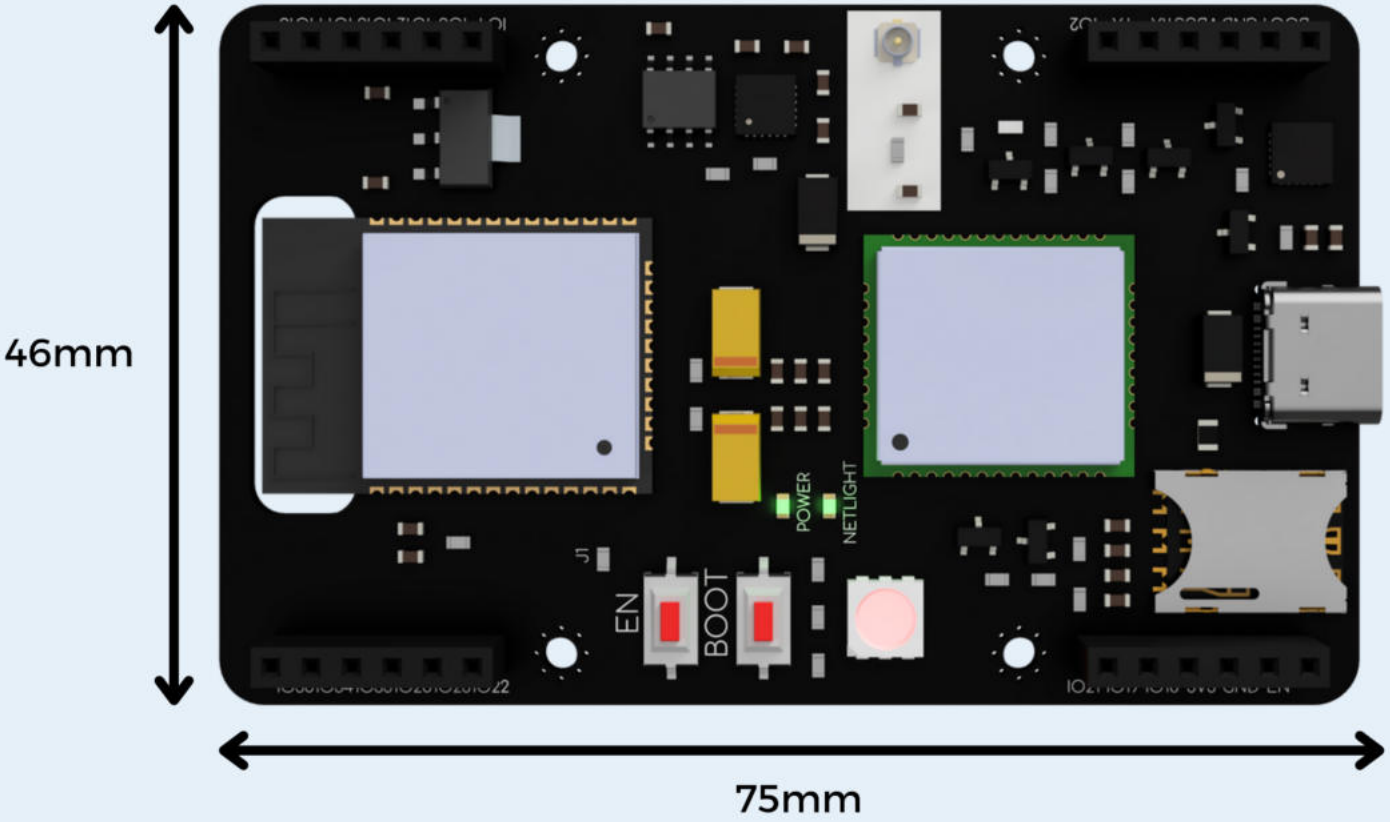


The Micromis Base V1 development board is also equipped with an addressable RGB LED to emit light signals. The mounted diode includes the WS2812C chip, which controls the diode and allows the user to select the color and color saturation for the diode's light. Due to the use of RGB technology, there are more than 16 million combinations at the user's disposal to achieve satisfying lighting effects.

The addressable LED is permanently connected to the 32 pin of the ESP32 chip and can be controlled using most libraries responsible for controlling addressable LEDs.

MICROMIS BASE V1: BOARD DIMENSIONS

The Micromis Base V1 platform, due to its compact size, can be used in a wide range of custom projects that require a control platform to be small in size while maintaining low power consumption, high performance and multiplatform communication via WiFi, Bluetooth or GSM.



Made by:
DEVICE PROTOTYPE

MICROMIS BASE V1: SAMPLE PROGRAMS - MODEM PRESENTATION

Using the Micromis Base V1 board is very easy due to the fact that the board is partially compatible with other popular solutions on the market, so we can confidently use programs for ESP32 itself, Quectel M65 modem, addressable diodes, IMU MPU6050, and LM75 temperature sensor. However, the DevicePrototype team has developed dedicated software for each additional component, so you can easily check how the components on your PCB work using the Arduino IDE environment.

The first program is "Modem presentation," which is a simple program that allows you to test the operation of the built-in modem. After uploading the program to the device and running Serial Monitor, we can type system commands that will control the modem and allow, for example, sending SMS messages, searching all available networks, configuring the modem or connecting to the network. Remember to complete the variables at the beginning of the program before uploading it, without them you won't be able to connect to the network and send SMS messages properly.

A very useful feature of this program is the ability to send AT commands to the modem. If you send some command that is not included in the list of supported commands then the program will automatically send it to the modem, this can greatly facilitate the work of slightly more advanced users who may want to build a scheme of sent commands to be added later to their own programs. The list of AT commands with their explanation is included in the board's resource packet and has been compiled by the modem manufacturer and divided into documents for each section of the modem's operation.

```
THE MODEM IS CONNECTED TO THE NETWORK:
Orange


NETWORKS SEARCHING
.....................................................................................................
AVAILABLE NETWORKS WITH DETAILS:
(2, Orange , Orange , 26003 ),(1, Plus , PLUS , 26001 ),(1, P4 , Play , 26006 ),(1, T-Mobile.pl , TM PL , 26002 ),,(0-4),(0-2)
```

```
DATA SENT BY USER: AT+COPS=?
DATA SENT BY MODEM:
AT+COPS=?
DATA SENT BY MODEM:

+COPS: (2,"Orange","Orange","26003"),(1,"Plus","PLUS","26001"),(1,"P4","Play","26006"),(1,"T-Mobile.pl","TM PL","26002"),,(0-4),(0-2)

OK
```

MICROMIS BASE V1: SAMPLE PROGRAMS - LED PRESENTATION

The second program is "LED presentation", it is a very short script that allows you to check the operation of the LED built into the Micromis Base V1 board. After uploading the program and running Serial Monitor, we have the option to send several commands to the LED, the commands can completely turn off the LED, set any color from the RGB palette or set one of the predetermined colors such as red, green, blue, pink, yellow or purple.

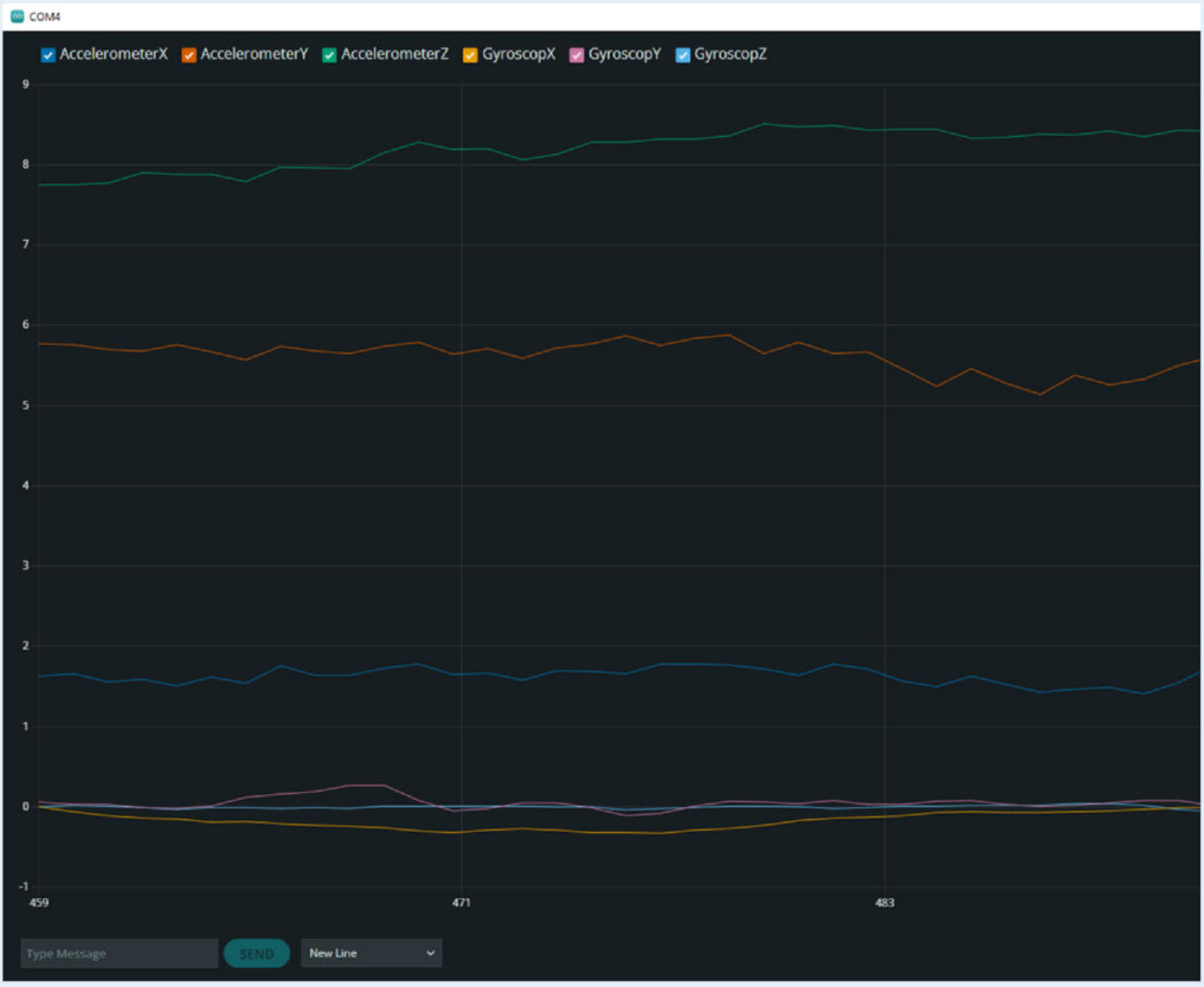
Based on the commands in the program code, novice users can easily build their own scripts to support the use of the addressable LED.

```
- GREEN - a command that sets the light color of the diode to green
- BLUE - a command that sets the light color of the diode to blue
- PINK - a command that sets the light color of the diode to pink
- YELLOW - a command that sets the light color of the diode to yellow
- PURPLE - a command that sets the light color of the diode to purple

--------------------------------------------------------------------------------

DATA SENT BY USER: RED

COLOR SET TO:255,0,0

DATA SENT BY USER: BLUE

COLOR SET TO:0,0,255

DATA SENT BY USER: YELLOW

COLOR SET TO:255,232,0

DATA SENT BY USER: CLEAR

COLOR SET TO: CLEAR

DATA SENT BY USER: PURPLE

COLOR SET TO:176,66,255

DATA SENT BY USER: COLOR:231,88,124

COLOR SET TO: 231,124,88
```

MICROMIS BASE V1: SAMPLE PROGRAMS - IMU PRESENTATION

The third program is "IMU Presentation", it is a very simple and short script that allows us to check how the IMU sensor embedded in the Micromis Base V1 board reads data. After uploading the program and running the Serial Plotter, we are able to view the data read from the IMU sensor in real time.

When you run Serial Plotter you can conveniently view the data that the board sends, every poke or movement of the board will be recorded and shown in graphs. Depending on your desire to check particular parameters, you can deselect individual measurement ranges to get information about only one particular data channel.



MICROMIS BASE V1: READY TO USE PROJECTS

To facilitate the use of Micromis Base V1 tiles, we have created a knowledge base that will allow you to access inspiring projects. We are constantly working on the content available on the website so you can easily check out sample applications of our products.

Don't wait and check it out now:
https://deviceprototype.com/hobby/knowledge-center/