

Architecture

Look at M.E.

Look at M.E. is built in a B-L072Z-LRWAN1 board in which we implemented an HC-SR04 ultrasonic proximity sensor to determine whether a visitor is in the “attention area” for each statue. Also we will use that same sensor as a trigger to change the lighting and sound of the room when several specific sensors are activated at the same time.

As for the software components, we created the main firmware file in Riot OS, the telemetries will be managed by the cloud-based broker Thingsboard and then retrieved on a web page to show all the important information to the museum curator.

The telemetries about the popularity of the art pieces and the lighting control will be sent via MQTT-SN through a PAHO transparent bridge to a Mosquitto broker and then forwarded to our cloud broker.

Let’s take a look at each component that makes our product starting with the hardware.

Hardware

B-L072Z-LRWAN1

The main hardware component of Look at M.E. is obviously the micro-controller board.

This is a B-L072Z-LRWAN1 which is a board manufactured by STMicroelectronics and is a pretty popular cause of its LoRa compatibility and its LoRaWAN class A certification.



The entire board's datasheet can be found at the following link:

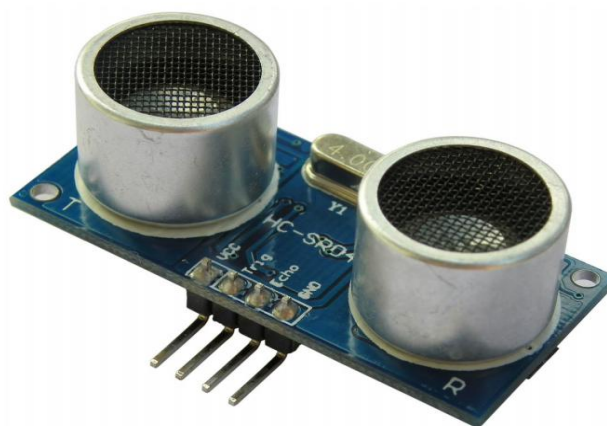
[Datasheet](#)

Others of its main features are:

- 7 LEDs:
 - 4 general-purpose LEDs
 - 5 V power LED
 - ST-LINK-communication LED
 - Fault-power LED
- 1 user and 1 reset push-buttons
- On-board ST-LINK/V2-1 debugger/programmer with USB re-enumeration capability: mass storage, Virtual COM port, and debug port
- Arduino™ Uno V3 connectors
- Board power supply through the USB bus or external VIN/3.3 V supply voltage or batteries
- 3 × AAA-type battery holder for standalone operation
- Support of a wide choice of Integrated Development Environments (IDEs) including IAR™, Keil®, GCC-based IDEs, Arm® Mbed™.

The HC-SR04 proximity sensor

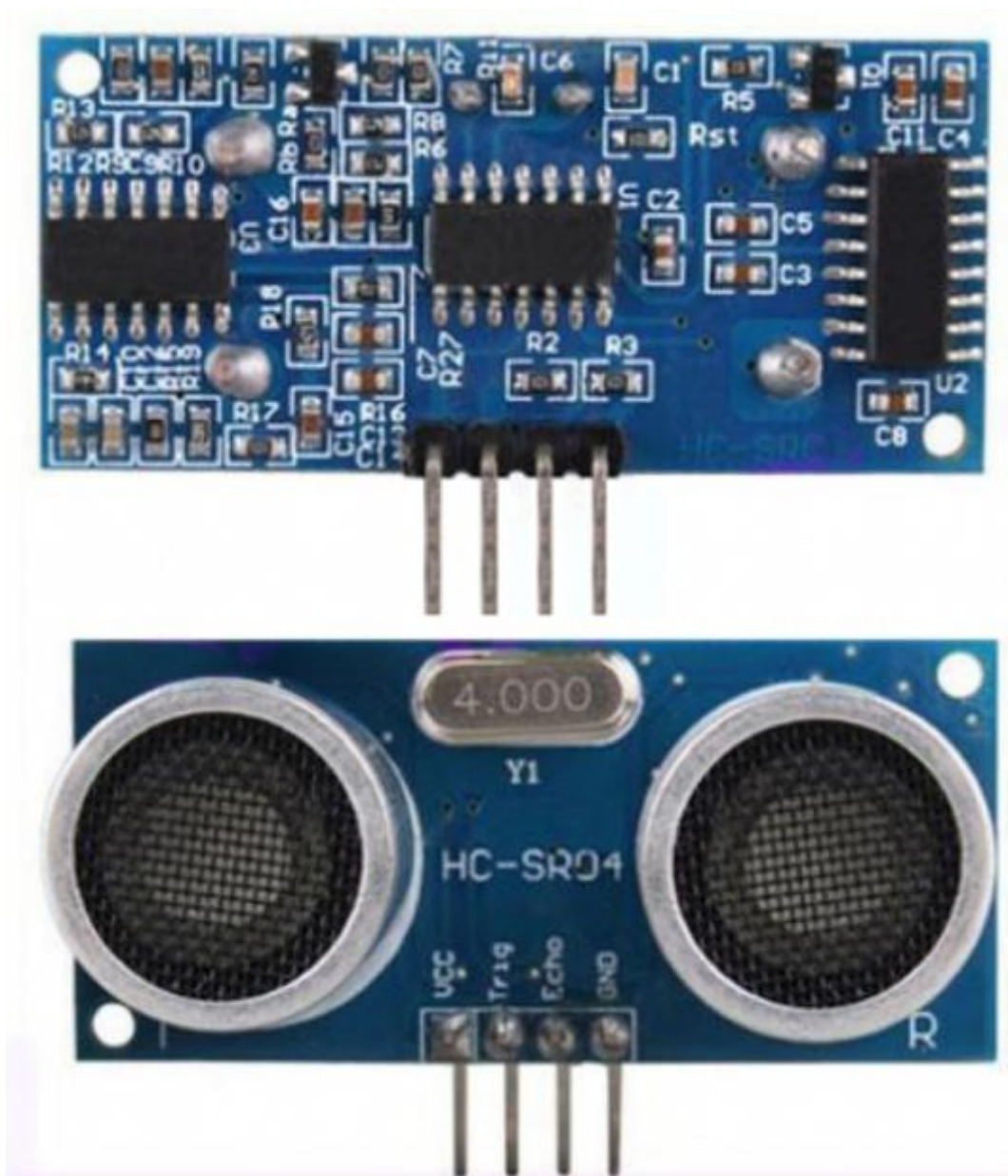
The HC-SR04 ultrasonic sensor uses sonar to determine distance to an object like bats or dolphins do. It offers excellent non-contact range detection with high accuracy and stable readings in an easy-to-use package. It operates in a distance range going from 2cm to 400 cm. Its operation is not affected by sunlight or black material like Sharp rangefinders are (although acoustically soft materials like cloth can be difficult to detect). It comes complete with an ultrasonic transmitter and receiver module.



The main features are :

Features:

- Power Supply :+5V DC
- Quiescent Current : <2mA
- Working Current: 15mA
- Effectual Angle: <15°
- Ranging Distance : 2cm – 400 cm
- Resolution : 0.3 cm
- Measuring Angle: 30 degree
- Trigger Input Pulse width: 10uS
- Dimension: 45mm x 20mm x 15mm



The sensor come with 4 pins that correspond to :

VCC = +5VDC

Trig = Trigger input of Sensor

Echo = Echo output of Sensor

GND = GND

To start the measurement, Trig of SR04 must receive a pulse of high (5V) for at least 10us, this will initiate the sensor will transmit out 8 cycle of ultrasonic burst at 40kHz and wait for the reflected ultrasonic burst. When the sensor detected ultrasonic from the receiver, it will set the Echo pin to high (5V) and delay for a period (width) which proportion to distance. To obtain the distance, measure the width (Ton) of the Echo pin.

External Hardware

Also, we have some hardware external to our board, for example the light and sound system that our application will use depending on the values of some variables in our script. These are external since we are not implementing them on the board but must be connected to our broker in order to receive the values that trigger them.

Software and Tools

The software components and tools of the product are a lot and different in their purposes; we have in fact different software to create the firmware and show collected data that uses libraries and APIs to access the sensor or publish telemetries.

Riot OS

The device will be created with RIOT OS, which is a real-time multi-threading operating system that supports a range of devices that are typically found in the Internet of Things. It allows us to use a C/C++ like language to manage all the logic of our product such as creating values, exclusive statements, or via the use of the emcute library, its possible for us to connect and publish to our broker.

Riot OS gives us the possibility to create the main file that can be flashed in a big range of boards or simulate the functioning of the software in the terminal.

The software Makefile, which is the place where we implement all the libraries we need for the compiling of our main file, will include modules for generic connection, IPv6 certification, the emcute module, and the hc-sr04 sensor module.

MQTT-SN

There is a recent increase of interest in Wireless Sensor Networks (WSNs), these networks can serve different purposes, from measurement and detection, to automation and process control.

MQTT protocol is an open and lightweight publish/subscribe protocol designed specifically for machine-to-machine communication and mobile applications. It is optimized for communications over networks where bandwidth is at a premium or where the network connection could be intermittent. However MQTT requires an underlying network, such as TCP/IP, that provides an ordered lossless connection capability and this is too complex for very simple, small footprint, and low-cost devices such as wireless SAs.

MQTT-SN can be considered as a version of MQTT which is adapted to the peculiarities of a wireless communication environment. In fact, it is designed to be as close as possible to MQTT, but is adapted to the peculiarities of a wireless communication environment such as low bandwidth, high link failures, short message length, etc.

Mosquitto RSMB

The Really Small Message Broker is a server implementation of the MQTT and MQTT-SN protocols. Any client that implements this protocol properly can use this server for sending and receiving messages.

As already said before, we will need the mosquitto RSMB because the clients that send data via MQTT-SN protocol cannot communicate directly to Thingsboard that receives MQTT messages. The broker will need a configuration file where we will specify all the needed variables to have the messages finally sent to our cloud broker.

Paho MQTT-SN Transparent Gateway

The MQTT-SN Transparent Gateway is a daemon, or small server, which accepts incoming MQTT-SN data over several transports (UDP, XBee) and converts it into MQTT appropriate for connecting to an MQTT server such as Eclipse Mosquitto. The Transparent gateway differs from the aggregating gateway that, as you can imagine, aggregates the MQTT-SN data into a single transmission over to the server. The transparent will leave the data “as it is” without any handling and forward them to the server.

Just like the RSMB, the gateway needs a configuration file where we can specify all the addresses and data we need in order to make it work as we need it to.

Thingsboard

ThingsBoard is an open-source IoT platform that enables rapid development, management and scaling of IoT projects. Its goal is to provide the out-of-the-box IoT cloud or on-premises solution that will enable server-side infrastructure for your IoT applications.

With ThingsBoard, you are able to:

- Provision devices, assets and customers and define relations between them.
- Collect and visualize data from devices and assets.
- Analyse incoming telemetry and trigger alarms with complex event processing.
- Control your devices using remote procedure calls (RPC).

- Build work-flows based on device life-cycle event, REST API event, RPC request, etc
- Design dynamic and responsive dashboards and present device or asset telemetry and insights to your customers
- Enable use-case specific features using customizable rule chains.
- Push device data to other systems.

This is the place where all the data we send will be stored, we can then use the received messages to trigger functions that will improve the user experience in the museum, or we can show them on a web page in a user-friendly format. In order to use the site, we will need a free account that can be created following the standard procedure.

Web Application

The web application will only have the purpose to show the popularity data to the museum curator so he can have a view on the most appreciated statues and act accordingly.

It will be a simple HTML page with some CSS style and some Javascript code that will manage the REST API from thingsboard to collect data from the Thingsboard database.

The web page will be accessible only by authorized staff so that the info stored in it can be kept safe.