**Code you need to edit**

The main tool we will use for this development is the RC-Switch library, which is open source and you can easily [download from GitHub](https://github.com/sui77/rc-switch/) or you can directly download our edited project [here](https://github.com/BehindTheSciences/Arduino-garage-door-opener).

**RCSwitch.cpp**

We have modified the protocol section:

#ifdef ESP8266

**static** **const** RCSwitch::Protocol proto[] = {

#else

**static** **const** RCSwitch::Protocol PROGMEM proto[] = {

#endif

  { 315, {  0, 0 }, {  1,  4 }, {  3,  2 }, **false** },    // protocol 1

  { 295, {  0, 0 }, {  1,  1 }, {  0,  1 }, **false** },    // protocol 2 preamble

  { 100, { 30, 71 }, {  4, 11 }, {  9,  6 }, **false** },    // protocol 3

  { 380, {  1,  6 }, {  1,  3 }, {  3,  1 }, **false** },    // protocol 4

  { 500, {  6, 14 }, {  1,  2 }, {  2,  1 }, **false** },    // protocol 5

  { 450, { 23,  1 }, {  1,  2 }, {  2,  1 }, **true** }      // protocol 6 (HT6P20B)

};

The meaning of the previous code is:

**Protocol 2** is the Preamble:

Here the short value is 295 us, so the digits are the times we multiple that pulse {low level, high level} by:

* Sync bit: {0,0}
* Zero bit: {1,1} first digit is a low level pulse, second digit is a high level pulse
* One bit: {0,1}

**Protocol 1** is the Data:

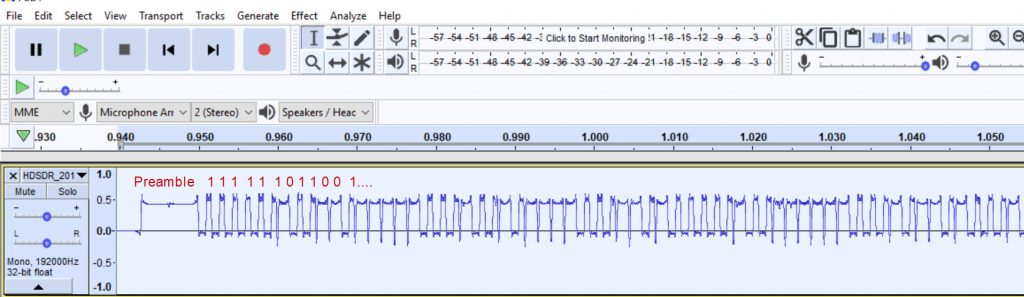
Here, 315 us is the short value

* Sync bit: {0,0}
* Zero bit: {1,4} =205 and 205×4 us first digit is a low level pulse, second digit is a high level pulse
* One bit: {3,2}

In order to get those pulse lengths, you will use the Matlab code provided here. We explain how to use it in the next section.

**Fob.ino**

You can find the pulse height by recording the fob signal with an RTL-SDR dongle and the free tool HDSDR, as shown [here](https://behindthesciences.com/tech-reviews/rtl-sdr-dongle-review/). Then, you can open that .wav file in Audacity (also free to download) and check:

Visual inspection with Audacity

As you can see in the previous image, we have written the corresponding 1s and 0s according to the pulse lengths we defined the protocols of RCSwitch.cpp. Once you have this information, you can edit the .ino code “send” section:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 | //Start Frame   //PreAmble  mySwitch.setProtocol(2);  mySwitch.send("111111111111111111111111");   //Data  mySwitch.setProtocol(1);  mySwitch.send("111111011001000001110110100001");  mySwitch.send("110010111110110000001110100101");  mySwitch.send("11010011010000100001"); |

The code above represents the fob signal we will transmit.  
The rest of the code, including the file to compile in Arduino and the Matlab code to find the pulse length are in our GitHub repository, so you can download them for free [here](https://github.com/BehindTheSciences/Arduino-garage-door-opener).

|  |
| --- |
| 詳細代碼  :  /\* |
|  | Example for different sending methods |
|  |  |
|  | https://github.com/sui77/rc-switch/ |
|  |  |
|  | \*/ |
|  |  |
|  | #include <RCSwitch.h> |
|  |  |
|  | RCSwitch mySwitch = RCSwitch(); |
|  | int PLength = 380; |
|  |  |
|  | void setup() { |
|  |  |
|  | Serial.begin(9600); |
|  |  |
|  | // Transmitter is connected to Arduino Pin #10 |
|  | mySwitch.enableTransmit(10); |
|  |  |
|  | // Optional set pulse length. |
|  | //mySwitch.setPulseLength(140); |
|  |  |
|  | // Optional set protocol (default is 1, will work for most outlets) |
|  | mySwitch.setProtocol(1); |
|  |  |
|  | // Optional set number of transmission repetitions. |
|  | mySwitch.setRepeatTransmit(1); |
|  |  |
|  | } |
|  |  |
|  | void loop() { |
|  |  |
|  | //Start Frame |
|  | //Start PreAmble |
|  | mySwitch.setProtocol(2); |
|  | mySwitch.send("111111111111111111111111"); |
|  | //digitalWrite(10,HIGH); |
|  | //delayMicroseconds(15000); |
|  | //Data |
|  | mySwitch.setProtocol(1); |
|  | mySwitch.send("111111011001000001110110100001"); |
|  | mySwitch.send("110010111110110000001110100101"); |
|  | mySwitch.send("11010011010000100001"); |
|  | // mySwitch.send("111000110000101110010011111000"); |
|  | // mySwitch.send("111001101011110111100111100010"); |
|  | // mySwitch.send("1110110111110111110001111100"); |
|  |  |
|  |  |
|  | delay(3); |
|  |  |
|  | } |