Getting Started

**Step 1: Download your raw Data from your remote.**

No programming required.

* Get an esp device, except the esp01.
* Go to VSCode with installed extension platfomio.
* click on home of VSCode (the little house on the bottom) , than, right of the funny orange Logo click on **New Project.**
* Give it a name. I called it IRecvDumpV3.  
  select your board from the list. I selected WeMos D1 R2 and mini (Wemos)  
  select the framework, (Arduino)  
  click finish  
  Now your project appears on the left side in the Workspace.
* Click on the platformio.ini file under your project.   
  the settings file opens.
* You see: **[env:d1\_mini]**, I changed this to **[env:esp8266]** because my serial monitor did not work. And also **board = D1\_mini** to **board=D1.** After the next upload everything worked.  
  *Here a more knowledgeable person needs to clarify this somehow. There is some info here:* [*https://docs.platformio.org/en/stable/boards/espressif8266/d1\_mini.html*](https://docs.platformio.org/en/stable/boards/espressif8266/d1_mini.html) *also mention monitor settings :* monitor\_rts = 0or1 monitor\_dtr = 0or1

Add under **framework = arduino** the line: **monitor\_speed = 115200**

* Click again on the home symbol on the bottom ( little house). In the vertical toolbar on the left side under the small orange logo, click on libraries. In the searchbar write: IRremoteESP8266, enter.  
  The first one popping up is the one from David Conran. Click on it, click **Add to Project,** select the project, click **Add.** Now look at your platformio.ini tab you will see , you got another line: **lib\_deps =** [**crankyoldgit/IRremoteESP8266@^2.8.6**](mailto:crankyoldgit/IRremoteESP8266@%5e2.8.6)
* On the left side in your workspace right click on **include. Or is it lib, please check?**select **New File…** Name it : **BaseOTA.h**Open Internet Browser and go to:[**https://github.com/crankyoldgit/IRremoteESP8266/tree/master/examples/IRrecvDumpV3/BaseOTA.h**](https://github.com/crankyoldgit/IRremoteESP8266/tree/master/examples/IRrecvDumpV3/BaseOTA.h)select the entire code, copy it and than paste it into your new file called BaseOTA.h
* Open Browser, go to:  
  [**https://github.com/crankyoldgit/IRremoteESP8266/tree/master/examples/IRrecvDumpV3/IRrecvDumpV3.ino**](https://github.com/crankyoldgit/IRremoteESP8266/tree/master/examples/IRrecvDumpV3/IRrecvDumpV3.ino)

Select the entire code, copy it and than go to your workspace and click on **src** and than on **main.ccp** , another tab opens. Inside you find the prewritten code: #include **<Arduino.h>** ….  
Delete the code that comes after it and paste the code you just copied from your internet browser.

* **That should be it, Done!**
* Now plug in your ESP to the USB Port and click on the arrow pointing to the right, that is on the bottom next to the home symbol, (the little house). If you get no error, it should upload your code to the esp.
* Than connect the IR sensor. On my esp the signal wire is connected to D5 (Pin 14)
* Connect your esp back to your computer, Go back to the bottom and click on the little symbol, that looks like a electrical plug. This will open the serial monitor.
* Now blast your esp with the infrared signal of the IR remote that came with your air-conditioner. The serial port should show you some information and the raw signal.  
  Mine shows me:  
  Timestamp : 000117.180 needs some info what it is  
  Library : v2.8.6 */the version of the library*Protocol : MIRAGE */the type of remote of your aircon*Code : 0x56750000200000000000000000002 (120 Bits) needs som info what it is and means  
  Mesg Desc.: Model: 2 (KKG29AC1) , …. */Type and name of the remote followed by the translated raw data into IR-Remote commands. F.e. Power: ON, Temp: 25C…..*  
  Than comes the raw Data consisting of many 3 and 4 digit numbers. It starts with:
* uint16\_t rawData[243] = {8362, 4252, 512, 558, …. ….  
  }; //MIRAGE  
  uint8\_t state[15] = {0x56, 0x75, 0x00, … … };   
  an short explanation for : u, int, 8, \_t, [15], 0x56, 8362, is needed here.

The most important word for me was MIRAGE. Now I know which libraries I have to use to build my little Automatic AC control that will use different settings for daylight and nighttime.

I guess, that will be Step two of my little adventure.

*Cheers, Lisa*