



1. Press [SW1](#) to bring up the **menu/confirm**
2. Press [SW3](#) & [SW4](#) to navigate **up/down**
3. Press [SW2](#) to **exit/cancel**

Arduino Setup

Watchy comes pre-loaded with firmware that demonstrates all the basic features. You can also try different watch faces and examples in Arduino.

1. Download and install the latest [Arduino IDE](#)
2. Start Arduino and open File > Preferences.
3. Under *Additional Board Manager URLs* add:

```
https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json
```

4. Open Tools > Board > **Boards Manager** and install the latest version of *esp32* platform
5. Under Sketch > Include Library > Manage Libraries, search for **Watchy** and install the latest version
6. Make sure all the dependencies are updated to the latest version i.e. **GxEPD2** , **WiFiManager**, **rtc_pcf8563**,etc.

Upload

1. Plug in the USB on Watchy and select the serial port that shows up

2. If nothing shows up, or if you're having trouble uploading, make sure you have the USB-Serial drivers installed. Also make sure you're using a USB data cable, and not a charge-only cable. Try different USB ports as well.
3. Select Tools > Board > ESP32 Arduino > **Watchy**
4. Select Tools > Board Revision > **Watchy v2.0**
5. Select Tools > Partition Scheme > **Huge App**
6. Leave everything else as default
7. Choose an example and click upload
8. Try modifying the examples or create your own app!

Wifi Setup

1. Click 'Setup Wifi' from main menu on the watch.
2. Connect to 'Watchy Ap' wifi from another device such as phone or computer / laptop.
3. Open 192.168.4.1 in the browser. On iphone this page open itself when connecting to wifi.
4. Click 'Configure WiFi'
5. Enter SSID and password. (Note: SSID can be prefilled by listed networks on the top of this screen)
6. Hit 'Save' button and wait for Watchy.
7. If connection failed, the watchy will display 'Setup failed & timed out!', otherwise it will display the local ip address and SSID of the connected network with confirmation.

Troubleshoot Wifi Setup:

1. Click 'Info' from 192.168.4.1 page.
2. Check your router setting to make sure the listed mac address is allowed.
3. If above didn't fix, then try clicking 'Erase Wifi Config' from 'Info' page and wait for Watchy to restart itself. Once it happen, try the wifi setup again and it should hopefully work.

PlatformIO Setup

PlatformIO is a compatible alternative to arduino. It's more oriented for the command line user, but it is also more flexible and predictable in build configurations and dependency management (like libraries).

It has two parts: a "core" that has the command line tools that build and upload/flash, and an "ide" which is a bunch of plugins and extensions for editors you can find [here](#).

Use whichever extensions you wish but this documentation is related to the core, so:

- Install with instructions [here](#).

Simple watchface example

This example is to create a new watch face project, it starts by copying one of the examples to the `src/` folder where you can make your own. However, it will not make it easy to edit the watchy library, or its `confia.h` file, which many want to, for that see the section below.

- Create a new folder and setup the PlatformIO project layout

```
mkdir my_new_watchy_face_project  
cd my_new_watchy_face_project
```

- Add the following to the `platformio.ini` file. Note that if you want to use another version of the Watchy library, you can put any file or git path here.

Some users have reported problems with one of the supported RTC modules: The module `PCF8563` seems to be supported during first boots, but their library is overridden by PlatformIO using a broken version - so you need to add another repository (`https://github.com/orbitalair/Rtc_Pcf8563.git`) to prevent that.

```
lib_deps =  
    sqfmi/Watchy ;  
    https://github.com/tzapu/WiFiManager.git#v2.0.11-beta ; Pinned  
for the same reason
```

- Also pin the version of platform `espressif32` to ensure compatibility.

```
[env:esp32dev]  
- platform = espressif32  
+ platform = espressif32 @ ~6.6.0
```

- Run PlatformIO, it will download dependencies such as the Watchy library, but then fail to compile because there aren't any source files in `src/` yet. So when the dependencies are downloaded, copy the `7_SE` example files to `src/`.

```
pio run # will fail compilation but will download dependencies  
cp .pio/libdeps/esp32dev/Watchy/examples/WatchFaces/7_SEQ/*.  
{ino,cpp,h?} src/
```

- Plug in your watchy, compile and then upload the watch face:

```
pio run -t upload
```

- Watch the serial port output

```
pio device monitor
```

- Celebrate by watching ascii star wars

```
telnet towel.blinkenlights.nl
```

- Additional keys you'll probably want to add to your `platformio.ini` file:

```
upload_speed = 3000000  
upload_port = /dev/cu.usbserial-MQK8G8  
monitor_port = /dev/cu.usbserial-MQK8G8  
monitor_speed = 115200
```