

# DDS Function Generator

## Production Programming: Pico W

Requirements:

1. PC running any major OS – process tested on Windows 11.
2. Arduino IDE V2.2.1 or later with Pico board and file upload utility.
3. USB micro cable.
4. Pico or Pico W to be programmed.

Major steps:

1. Prepare the Arduino environment
2. Upload main program.
3. Upload required webserver files to a LittleFS partition using file upload capability in Arduino IDE.
4. Validate program.

## Arduino method (no WiFi required)

### Prepare Arduino environment

Note: Arduino IDE later than 2.2.1 is required for the “Upload LittleFS to Pico/ESP32” plug in to function.

1. Add an additional board URL (Preferences menu) for the Pico  
[https://github.com/earlephilhower/arduino-pico/releases/download/global/package\\_rp2040\\_index.json](https://github.com/earlephilhower/arduino-pico/releases/download/global/package_rp2040_index.json)
2. Install the Raspberry Pi Pic/RP2040 (Earle Philhower) board using the Arduino Boards Manager.  
<https://github.com/earlephilhower/arduino-pico>
3. Add the LittleFS uploader utility  
<https://github.com/earlephilhower/arduino-littlefs-upload>  
see last paragraph on main Github page for installation instructions
4. Use a dummy Arduino sketch, with files in a “data” subfolder, and the “Upload LittleFS to Pico/ESP32” plug-in tool to Pico/ESP32” to upload files.
  1. Create a dummy Sketch.
  2. Set the Tools > Board to Raspberry Pi Pico/RP2040 > Raspberry Pi Pico W.
  3. Set the Flash Size to “2MB (Sketch 1920KB, FS 128KB)”
  4. The CPU speed and other parameters do not matter, as they are encoded into the program’s UF2 file.
  5. The COM Port will be set after uploading the main program.
  6. Save the sketch. (once saved, the IDE will remember the Board’s parameters)
  7. Put any files to be uploaded into a “data” subfolder of your sketch.

Board: "Raspberry Pi Pico W"
Port: "COM6"
Get Board Info
Debug Level: "None"
Debug Port: "Disabled"
C++ Exceptions: "Disabled"
Flash Size: "2MB (Sketch: 1920KB, FS: 128KB)"
CPU Speed: "133 MHz"
IP/Bluetooth Stack: "IPv4 Only"
Optimize: "Small (-Os) (standard)"
RTTI: "Disabled"
Stack Protector: "Disabled"
Upload Method: "Default (UF2)"
USB Stack: "Pico SDK"
WiFi Region: "Worldwide"

## For each Pico to be programmed:

### 1. Upload main program

1. Connect the Pico while holding its white button down, to put it into programming mode.  
(Holding down the button is not necessary with virgin Picos)
2. A drive should appear on your host OS with a name something like RPI-RP2 in the computer's file manager.
3. Copy UF2 program to the virtual drive using the computer's file management tool.
4. When the virtual drive disappears, the Pico has completed the programming cycle.

### 2. Upload additional webserver files

1. In the Arduino IDE, open the dummy sketch prepared above, with the files to be uploaded in a /data subfolder.
2. Reconnect the Pico and set the COM port (it may change for each Pico being programmed).
3. Close the Serial Monitor, if it is open, as uploading will fail otherwise.
4. Ctrl-Shift-P will bring up the Arduino shortcuts menu.
5. Type in "Upload" to locate the "Upload LittleFS to Pico/ESP32" plug in and execute it.
6. Files from the "data" sub folder will copy onto the Pico and it will automatically reset.
7. File uploading is complete.

### 3. Validation

Programming can be validated by connecting it to the Arduino IDE and checking the POST messages on the Arduino's Serial Terminal function after selecting the appropriate COM port.

1. Open the Serial Monitor with the COM port set above.
2. Reconnect the Pico.
3. Output something like the following should occur as the Pico boots:

```
--- DDS Function Generator ---  
Board version 3  
EEPROM (major) version 1  
Software (minor) version 1  
Starting TFT  
No EEPROM found
```

If the web files haven't been uploaded then the POST message will include:

```
Missing web files
```