Creative Embedded Systems: Lab 4

Welcome to analog

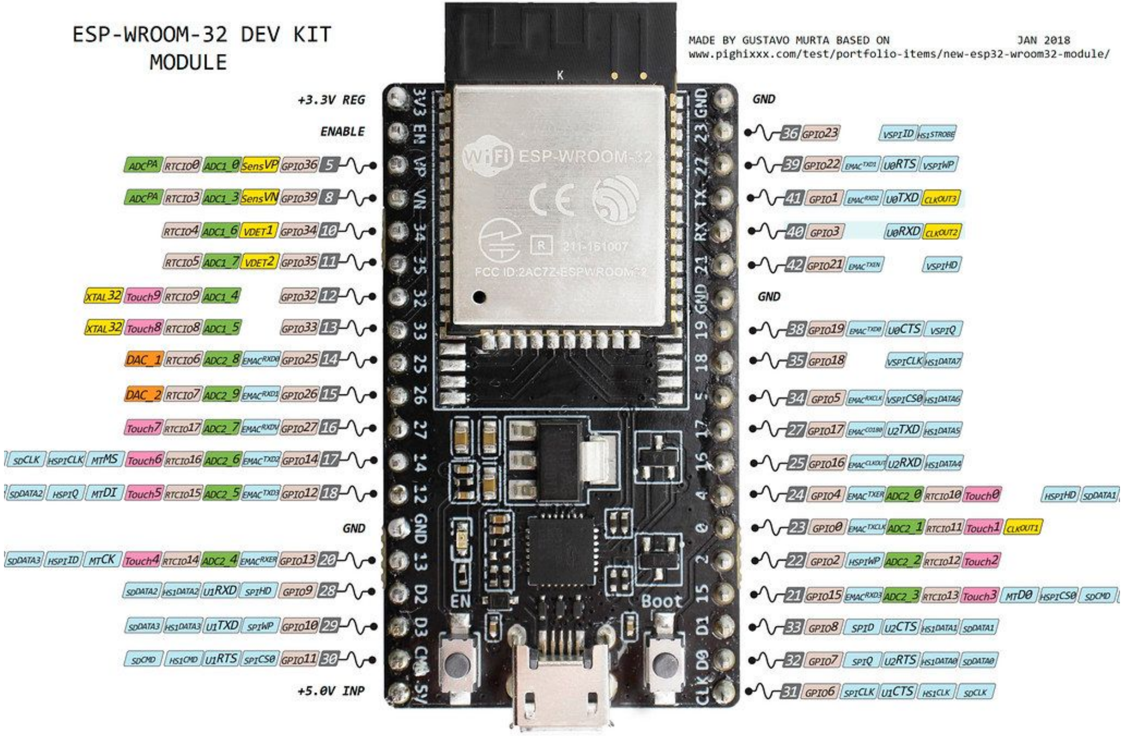
Announcements

- Repos. Public. End of discussion.
- Cables!

ESP32 (ESP-WROOM 32D to be specific)

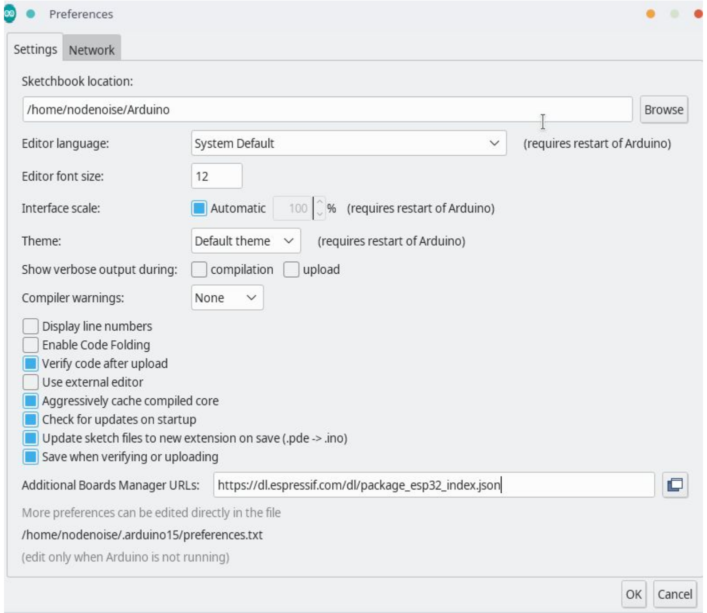
- 18 Analog-to-Digital Converter (ADC) channels
- 3 SPI interfaces
- 3 UART interfaces
- 2 I2C interfaces
- 16 PWM output channels
- 2 Digital-to-Analog Converters (DAC)
- 2 I2S interfaces
- 10 Capacitive sensing GPIOs

Can be programmed through Arduino with the ESP32 Library


ESP-WROOM-32 DEV KIT
MODULE
MADE BY GUSTAVO MURTA BASED ON JAN 2018
www.pighixxx.com/test/portfolio-items/new-esp32-wroom32-module/
+3.3V REG
ENABLE
ADC PA RTCIO0 ADC1_0 SensVP GPIO36 5
ADC PA RTCIO3 ADC1_3 SensVN GPIO39 8
RTCIO4 ADC1_6 VDET1 GPIO34 10
RTCIO5 ADC1_7 VDET2 GPIO35 11
XTAL32 Touch9 RTCIO9 ADC1_4 GPIO32 12
XTAL32 Touch8 RTCIO8 ADC1_5 GPIO33 13
DAC_1 RTCIO6 ADC2_8 EMAC RXD0 GPIO25 14
DAC_2 RTCIO7 ADC2_9 EMAC RXD1 GPIO26 15
Touch7 RTCIO17 ADC2_7 EMAC RXDV GPIO27 16
SDCLK HSPICLK MTMS Touch6 RTCIO16 ADC2_6 EMAC TXD2 GPIO14 17
SDDATA2 HSPIQ MTDI Touch5 RTCIO15 ADC2_5 EMAC TXD3 GPIO12 18
GND
SDDATA3 HSPIID MTCK Touch4 RTCIO14 ADC2_4 EMAC RXER GPIO13 20
SDDATA2 HS1DATA2 U1RXD SPIHD GPIO9 28
SDDATA3 HS1DATA3 U1TXD SPIWP GPIO10 29
SDCMD HS1CMD U1RTS SPICS0 GPIO11 30
+5.0V INP
3V3 EN VP VN 34 35 32 33 25 26 27 14 12 GND 13 D2 D3 CMD 5V
GND 23 22 TX RX 21 GND 19 18 5 17 16 4 0 2 15 D1 D0 CLK
WiFi ESP-WROOM-32
FCC ID:2AC7Z-ESPWROOM32
211-161007
EN
Boot
GND
36 GPIO23 VSPIID HS1STROBE
39 GPIO22 EMAC TXD1 U0RTS VSPIWP
41 GPIO1 EMAC RXD2 U0TXD CLKOUT3
40 GPIO3 U0RXD CLKOUT2
42 GPIO21 EMAC TXEN VSPIHD
GND
38 GPIO19 EMAC TXD0 U0CTS VSPIQ
35 GPIO18 VSPICLK HS1DATA7
34 GPIO5 EMAC RXCLK VSPICS0 HS1DATA6
27 GPIO17 EMAC CO180 U2TXD HS1DATA5
25 GPIO16 EMAC CLKOUT U2RXD HS1DATA4
24 GPIO4 EMAC TXER ADC2_0 RTCIO10 Touch0 HSPIHD SDDATA1
23 GPIO0 EMAC TXCLK ADC2_1 RTCIO11 Touch1 CLKOUT1
22 GPIO2 HSPIWP ADC2_2 RTCIO12 Touch2
21 GPIO15 EMAC RXD3 ADC2_3 RTCIO13 Touch3 MTDO HSPICS0 SDCMD
33 GPIO8 SPID U2CTS HS1DATA1 SDDATA1
32 GPIO7 SPIQ U2RTS HS1DATA0 SDDATA0
31 GPIO6 SPICLK U1CTS HS1CLK SDCLK

ESP32 Programming: Arduino IDE Setup

You can use almost anything to program the ESP32, but the Arduino IDE is probably easiest for a number of reasons.



Sketchbook location:
/home/nodenoise/Arduino

Editor language: System Default (requires restart of Arduino)

Editor font size: 12

Interface scale: ☑ Automatic 100 % (requires restart of Arduino)

Theme: Default theme (requires restart of Arduino)

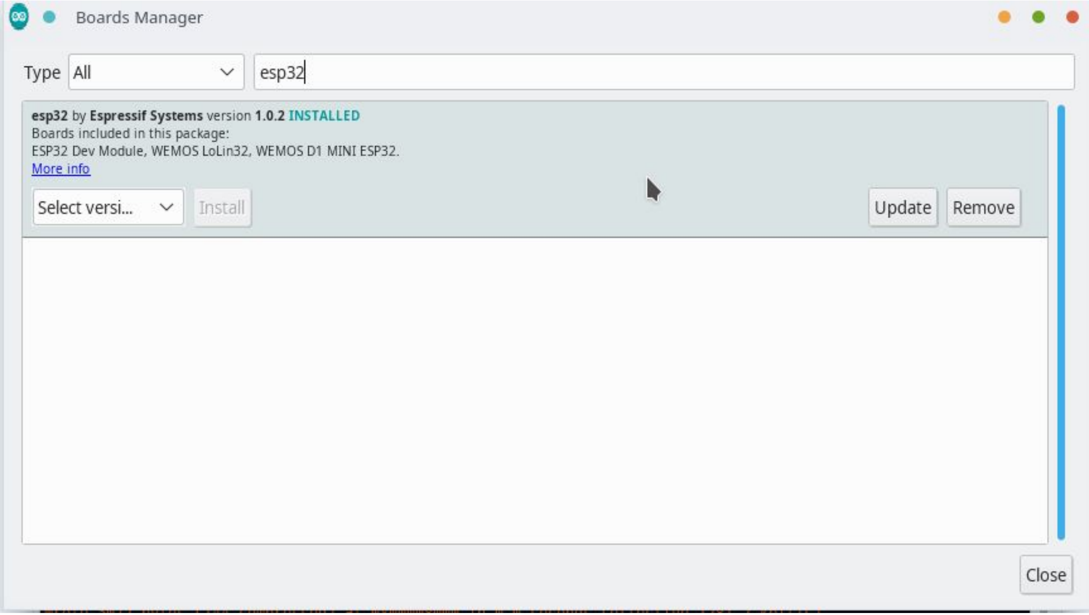
Show verbose output during: ☐ compilation ☐ upload

Compiler warnings: None

☐ Display line numbers
☐ Enable Code Folding
☑ Verify code after upload
☐ Use external editor
☑ Aggressively cache compiled core
☑ Check for updates on startup
☑ Update sketch files to new extension on save (.pde -> .ino)
☑ Save when verifying or uploading

Additional Boards Manager URLs: https://dl.espressif.com/dl/package_esp32_index.json

More preferences can be edited directly in the file
/home/nodenoise/.arduino15/preferences.txt
(edit only when Arduino is not running)



Type All esp32

esp32 by **Espressif Systems** version **1.0.2** **INSTALLED**
Boards included in this package:
ESP32 Dev Module, WEMOS LoLin32, WEMOS D1 MINI ESP32.
More info

ESP32 Programming: Arduino IDE Setup

Espressif Docs:
https://docs.espressif.com/projects/esp-idf/en/latest/index.html

API:
https://docs.espressif.com/projects/esp-idf/en/latest/api-reference/peripherals/index.html

Datasheet:
https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32d_esp32-wroom-32u_datasheet_en.pdf

Power Considerations for ESP32

USB (5V): A device may draw a maximum of 5 unit loads (500 mA) from a port in USB 2.0; 6 (900 mA) in USB 3.0.

GPIO, 5V+GND: can take up to 12V, but don't. Optimal seems to be 7V ish to limit current draw.

GPIO, 3V3+GND: ONLY 3.3V in (with no internal regulation). NOT RECOMMENDED.

Section 5: Electrical Characteristics
https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32d_esp32-wroom-32u_datasheet_en.pdf

Plain English Expo:
https://techexplorations.com/guides/esp32/begin/power/

ESP32-->Pi Communication

First: check your groups on Pi:

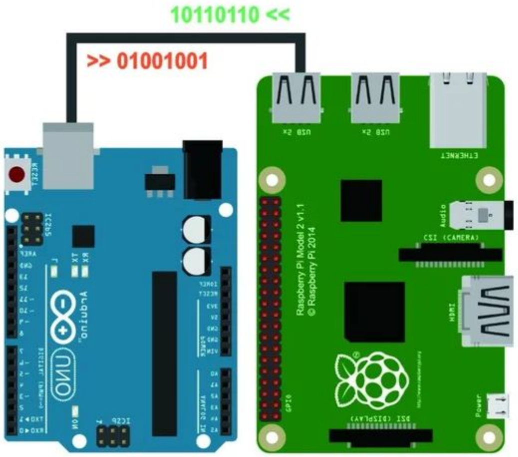
```
$ groups
```

If not in dialout group, add:

```
$ sudo usermod -a -G dialout pi
```

Plug ESP32 microUSB → Pi Type A

Include Serial.print() and/or Serial.println() to communicate data to the Pi



https://docs.espressif.com/projects/esp-idf/en/latest/get-started/establish-serial-connection.html

Example with built-in sensor:
https://microcontrollerslab.com/esp32-built-in-hall-effect-sensor-with-arduino-ide/

Lab Practicum: Pair programming the ESP32 to communicate with Raspi

1. Setup Arduino IDE on Laptop
2. Program ESP32 (DevkitC) for:
    a. Joystick: analog input
    b. Button: digital input
    c. Switch: digital input
3. Communicate readings to Raspi via Serial
    a. Raspi should run python script to read values from the USB port (e.g. pyserial library)