

Introduction to ESP32 using Vaman

G V V Sharma*

Contents

1	Software	1
2	Hardware Setup	1
3	Blink LED	1
4	ESP IDF	2
5	Raspberry Pi	2
5.1	Enable Serial Communication	2
5.2	Flash Vaman-ESP	2

Abstract—This document provides a simple introduction to programming the ESP32 on Vaman using the Arduino framework

1 Software

All codes used in this document are available at the following link

<https://github.com/gadepall/vaman/tree/master/esp32/setup/codes/>

2 Hardware Setup

2.1. Connect the USB-UART to raspberry pi through USB.

2.2. On the rpi

```
dmesg | tail
lsusb
```

you should see the USB-UART connector detected.

2.3. Connect the USB-UART pins to the Vaman ESP32 pins according to Table 2.3.1

2.4. Connect the Vaman-ESP pins to the seven segment display according to Table 2.4.1
The GPIO pins are listed in Table 2.4.2
Note that these pins can be used for sev-

VAMAN LC PINS	UART PINS
GND	GND
ENB	ENB
TXD0	RXD
RXD0	TXD
0	IO0
5V	5V

TABLE 2.3.1

ESP	SEVEN SEGMENT DISPLAY
5V	COM
2	DOT

TABLE 2.4.1

eral functions, refer to the ESP32 datasheet for details. The Vaman pin diagram is available in Fig. 2.4.1

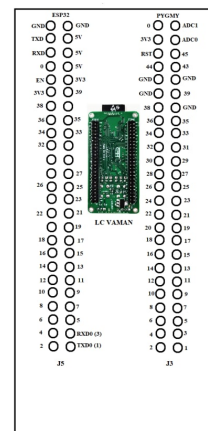


Fig. 2.4.1

3 Blink LED

3.1. On termux on your phone,

```
svn co https://github.com/gadepall/vaman/trunk/esp32/setup/codes/ide/blink
```

*The author is with the Department of Electrical Engineering, IIT Hyderabad, 502285. email: gadepall@ee.iith.ac.in! All content in this manual is released under GNU/GPL.

GPIO	Input	Others
2	34	1
4	35	3
5	36	6
10	37	7
12	38	8
13	39	9
14		10
15		11
16		
17		
18		
19		
21		
22		
23		
25		
26		
27		
32		
33		

TABLE 2.4.2

```
cd blink
pio run
```

3.2. Transfer the ini and bin files to the rpi

```
scp platformio.ini pi@192
.168.50.252:./hi/platformio.
ini

scp .pio/build/esp32doit-devkit
-v1/firmware.bin pi@192
.168.50.252:./hi/.pio/build/
esp32doit-devkit-v1/firmware
.bin
```

3.3. On rpi,

```
cd /home/pi/hi
pio run -t nobuild -t upload
```

3.4. On your phone, open

```
src/main.cpp
```

and change the delay to

```
delay(2000);
```

and execute the code by following the steps above.

4 ESP IDF

- 4.1. Earlier, we were using the arduino framework, where the programming language was arduino. In the following directory, the same functionality is achieved through a C program.

```
svn co https://github.com/
gadepall/vaman/trunk/esp32/
setup/codes/idf/blink
cd blink
pio run
```

- 4.2. The flashing process remains the same.

5 Raspberry Pi

5.1 Enable Serial Communication

5.1.1. On the RPi,

```
sudo raspi-config
```

5.1.2. Select Interfacing Options

5.1.3. Then select Serial Port

5.1.4. Reply no to login shell over serial

5.1.5. Say yes to running hardware over serial port.

5.1.6. Connect the rpi tx (pin 8) and rx (pin 10)

5.1.7. Install minicom and start it

```
sudo apt install minicom
minicom -b 115200 -o -D /dev/
serial0
```

Type namaste. If you see it displayed on screen, your serial port is working.

5.2 Flash Vaman-ESP

- 5.2.1. Since the RPi supports UART through its GPIO pins, a separate USB UART adapter is not required. Table 2.3.1 can be accordingly modified to obtain Table 5.2.1.1 On RPi, the pin numbers for serial communication are Tx=8, Rx=10.

- 5.2.2. Modify your platformio.ini file by adding the line

VAMAN LC PINS	RPI/UART PINS
GND	GND
ENB	GND
TXD0	RXD
RXD0	TXD
0	GND
5V	5V

TABLE 5.2.1.1

```
upload_port = /dev/serial0
```

5.2.3. After executing

```
pio run -t nobuild -t upload
```

while the dots and dashes are printed on the screen, disconnect the EN wire from GND. The Vaman-ESP should now flash.

5.2.4. After flashing, disconnect pin 0 on Vaman-ESP from GND. Power on Vaman and the appropriate LED will blink.