EE 49
Electronics for IoT

Microcontroller Unit
Computer
More Computers
Microcontroller

A complete computer on a “chip”!
Transistor
Microcontroller Blockdiagram
MCU

• Microcontroller Unit consists of
  – Datapath (compute)
  – Memory
  – IO

• MCU
EE49 MCU
Huzzah32 “Breakout” Board
Espressif – ESP32

ESP32
A Different IoT
Power and
Performance
ESP32 Datasheet

1 Overview
1.1 Featured Solutions
1.1.1 Ultra-Low-Power Solution
1.1.2 Complete Integration Solution
1.2 Wi-Fi Key Features
1.3 BT Key Features
1.4 MCU and Advanced Features
1.4.1 CPU and Memory
1.4.2 Clocks and Timers
1.4.3 Advanced Peripheral Interfaces
1.4.4 Security
1.5 Applications (A Non-exhaustive List)
1.6 Block Diagram

2 Pin Definitions
2.1 Pin Layout
2.2 Pin Description
2.3 Power Scheme
2.4 Strapping Pins

3 Functional Description
3.1 CPU and Memory
3.1.1 CPU
3.1.2 Internal Memory
3.1.3 External Flash and SRAM
3.1.4 Memory Map
3.2 Timers and Watchdogs
3.2.1 64-bit Timers
3.2.2 Watchdog Timers
3.3 System Clocks
3.3.1 CPU Clock
3.3.2 RTC Clock
3.3.3 Audio PLL Clock
3.4 Radio
3.4.1 2.4 GHz Receiver
3.4.2 2.4 GHz Transmitter
3.4.3 Clock Generator
3.5 Wi-Fi
3.5.1 Wi-Fi Radio and Baseband
3.5.2 Wi-Fi MAC
3.6 Bluetooth
3.6.1 Bluetooth Radio and Baseband
3.6.2 Bluetooth Interface
3.6.3 Bluetooth Stack

3.6.4 Bluetooth Link Controller
3.7 RTC and Low-Power Management

4 Peripherals and Sensors
4.1 Descriptions of Peripherals and Sensors
4.1.1 General Purpose Input / Output Interface (GPIO)
4.1.2 Analog-to-Digital Converter (ADC)
4.1.3 Hall Sensor
4.1.4 Digital-to-Analog Converter (DAC)
4.1.5 Touch Sensor
4.1.6 Ultra-Lower-Power Co-processor
4.1.7 Ethernet MAC Interface
4.1.8 SD/SDIO/MMC Host Controller
4.1.9 SDIO/SPI Slave Controller
4.1.10 Universal Asynchronous Receiver Transmitter (UART)
4.1.11 RS232 Interface
4.1.12 PS Interface
4.1.13 Infrared Remote Controller
4.1.14 Pulse Counter
4.1.15 Pulse Width Modulation (PWM)
4.1.16 LED PWM
4.1.17 Serial Peripheral Interface (SPI)
4.1.18 Accelerator
4.2 Peripheral Pin Configurations

5 Electrical Characteristics
5.1 Absolute Maximum Ratings
5.2 Recommended Operating Conditions
5.3 DC Characteristics (3.3 V, 25 °C)
5.4 Reliability Qualifications
5.5 RF Power-Consumption Specifications
5.6 Wi-Fi Radio
5.7 Bluetooth Radio
5.7.1 Receiver – Basic Data Rate
5.7.2 Transmitter – Basic Data Rate
5.7.3 Receiver – Enhanced Data Rate
5.7.4 Transmitter – Enhanced Data Rate
5.8 Bluetooth LE Radio
5.8.1 Receiver
5.8.2 Transmitter

6 Package Information

7 Part Number and Ordering Information

8 Learning Resources
8.1 Must-Read Documents
## Memory

<table>
<thead>
<tr>
<th>Random Access Memory (RAM)</th>
<th>Flash Memory</th>
</tr>
</thead>
</table>

B. E. Boser

IoT49: Microcontroller
Bit
Binary Number
Byte, Word
## Binary vs Decimal

<table>
<thead>
<tr>
<th>Decimal Number</th>
<th>Binary Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2 = 2^1</td>
<td>10</td>
</tr>
<tr>
<td>3</td>
<td>11</td>
</tr>
<tr>
<td>4 = 2^2</td>
<td>100</td>
</tr>
<tr>
<td>5</td>
<td>101</td>
</tr>
<tr>
<td>6</td>
<td>110</td>
</tr>
<tr>
<td>7</td>
<td>111</td>
</tr>
<tr>
<td>8 = 2^3</td>
<td>1000</td>
</tr>
<tr>
<td>9</td>
<td>1001</td>
</tr>
<tr>
<td>10</td>
<td>1010</td>
</tr>
<tr>
<td>11</td>
<td>1011</td>
</tr>
<tr>
<td>12</td>
<td>1100</td>
</tr>
<tr>
<td>13</td>
<td>1101</td>
</tr>
<tr>
<td>14</td>
<td>1110</td>
</tr>
<tr>
<td>15</td>
<td>1111</td>
</tr>
<tr>
<td>16 = 2^4</td>
<td>10000</td>
</tr>
<tr>
<td>17</td>
<td>10001</td>
</tr>
<tr>
<td>18</td>
<td>10010</td>
</tr>
<tr>
<td>19</td>
<td>10011</td>
</tr>
<tr>
<td>20</td>
<td>10100</td>
</tr>
<tr>
<td>21</td>
<td>10101</td>
</tr>
<tr>
<td>22</td>
<td>10110</td>
</tr>
<tr>
<td>23</td>
<td>10111</td>
</tr>
<tr>
<td>24</td>
<td>11000</td>
</tr>
<tr>
<td>25</td>
<td>11011</td>
</tr>
<tr>
<td>26</td>
<td>11100</td>
</tr>
<tr>
<td>27</td>
<td>11101</td>
</tr>
<tr>
<td>28</td>
<td>11110</td>
</tr>
<tr>
<td>29</td>
<td>11111</td>
</tr>
<tr>
<td>30</td>
<td>11111</td>
</tr>
<tr>
<td>31</td>
<td>11111</td>
</tr>
<tr>
<td>32 = 2^5</td>
<td>100000</td>
</tr>
</tbody>
</table>
Conversion

1 + 8 + 16 + 64 + 128 = 217

https://www.tutorialspoint.com/computer_fundamentals/computer_number_conversion.htm
Hexadecimal

$890234_{10} = 11011001010101111010_2$
Programming
MicroPython

- Python 3.4 syntax
- Same “core” libraries (e.g. math functions)
- Different specialty libraries
  - Talk to sensors
  - No functions for graphical user interface
Programming Setup
1st Step: “Flash” MicroPython

ESP32 MicroPython Installation

This guide shows how to install MicroPython on an ESP32 microcontroller (Adafruit Huzzah32 used here, but should work with other boards also) and programming it with it with the help of Jupyter Notebooks.

What’s a Jupyter Notebook? You are looking at one! A mix of cells containing code, explanations, images, etc. With the correct setup the Notebook is “alive”, i.e. you can execute and modify the code in the notebook.

**Download the ESP32 Repository**

First you need to download these instructions and related files. To do this, you need to install git on your computer. Follow the instructions on [https://git-scm.com](https://git-scm.com).

Now open a terminal window on your computer (e.g. `cmd.exe` on Windows or `terminal.app` on macOS) in the folder where you want to save your esp32 programs and run the following command:

```bash
  git clone https://github.com/bboser/esp32.git
```

**Install Python 3**