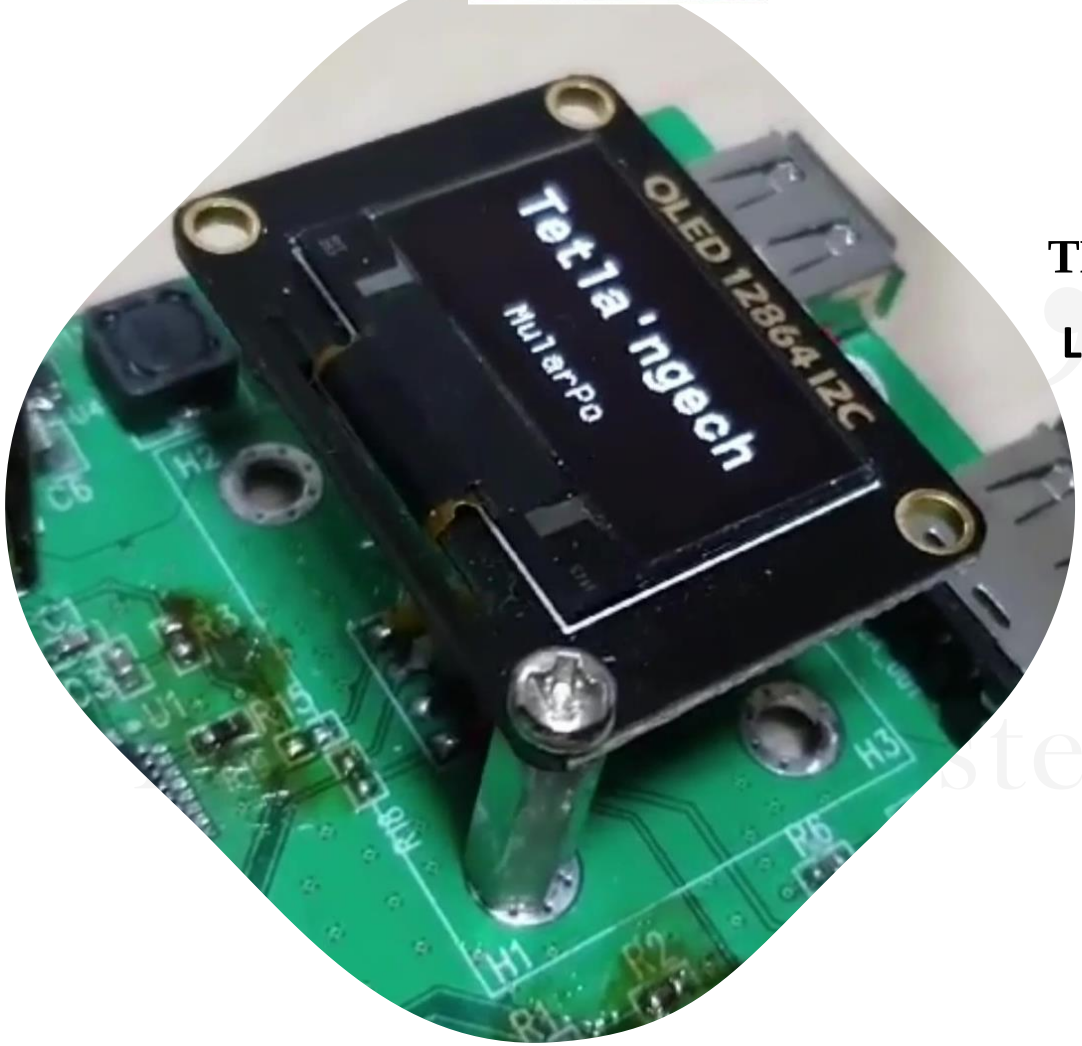
KYAMBOGO UNIVERSITY

Tet la'ngech

Embedded Systems Technology



TETE 3102 MICROCONTROLLERS AND APPLICATIONS

Lecturer: Dickson Mugerwa (Ph.D.)

Embedded Systems

Hardware in a nutshell

Special Guest (Presenter)

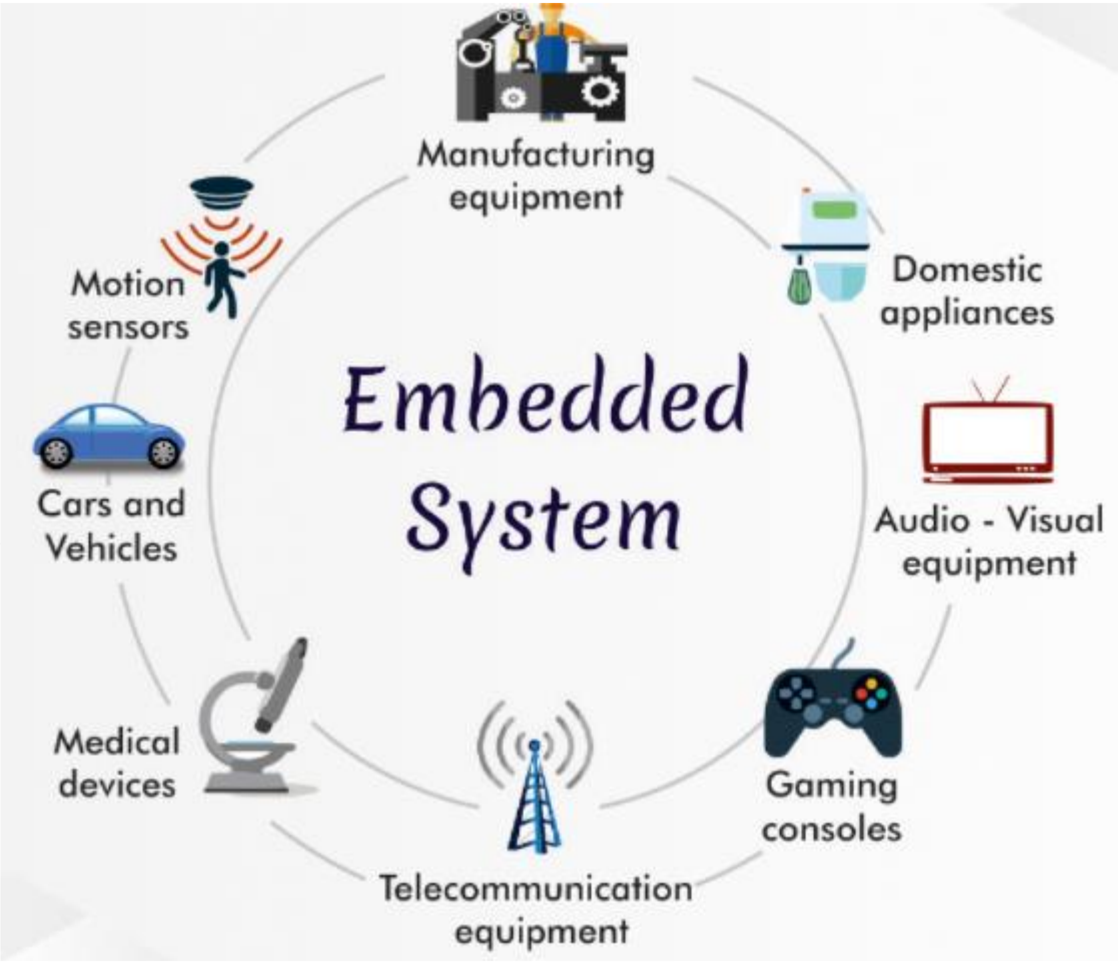
By; Muwakanya Khasim J

Chief Executive Officer

"Transforming Uganda through advanced embedded systems technology"

Embedded Systems

- **Product Design** — General Outlook of the system
- **Hardware** — Components that run the whole system
- **Software** — Code that provides Interaction between user and product (UI)
- **Firmware** — Code that runs the components



Embedded Systems

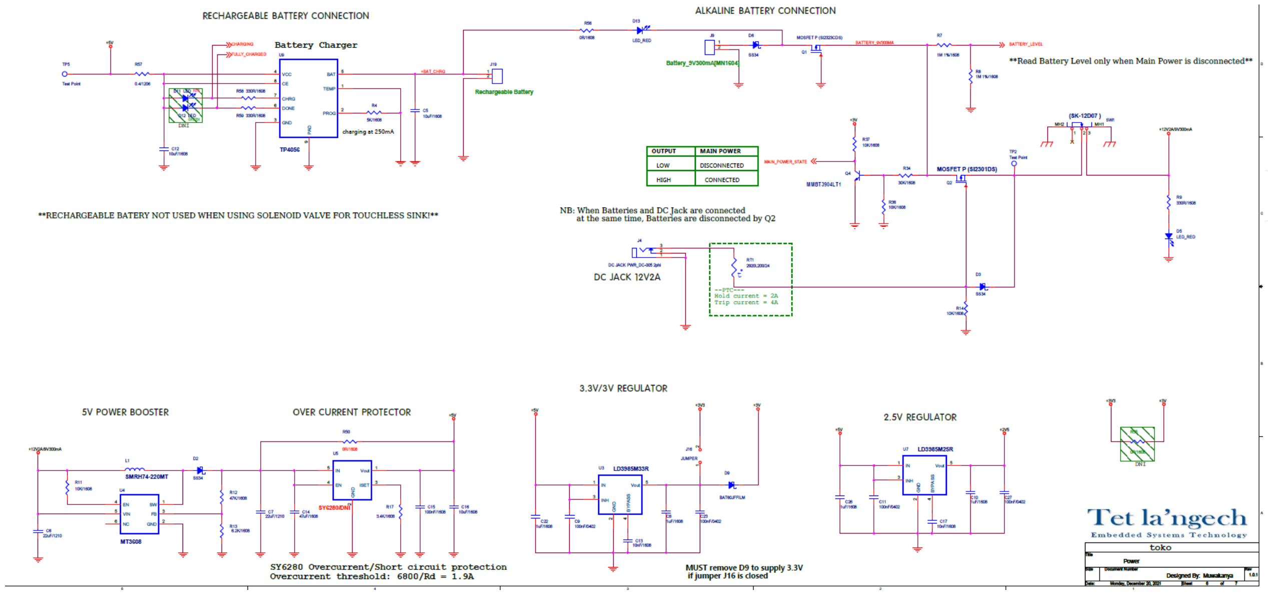
Hardware

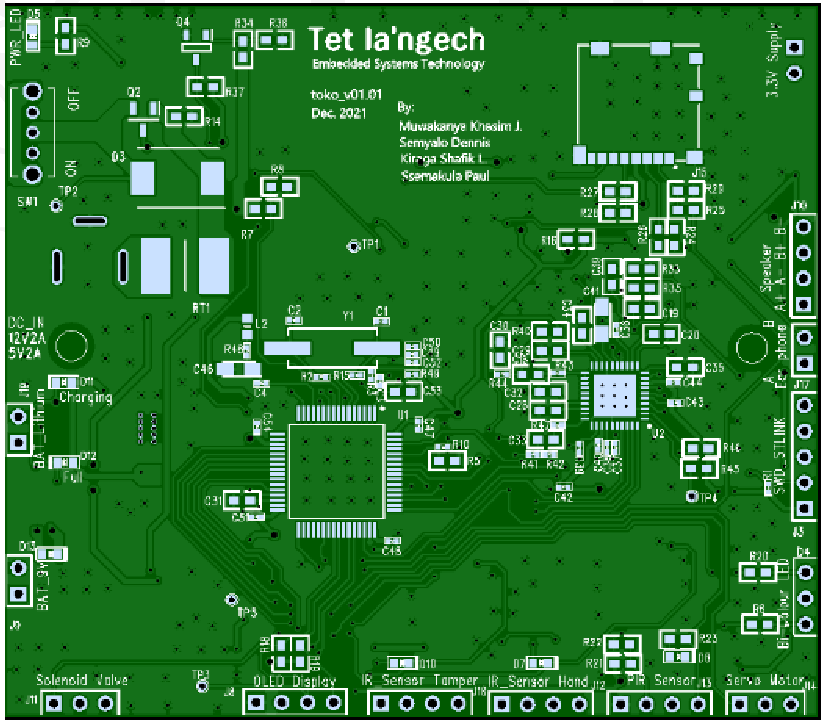
Components that run the whole system

Board Designing

Schematics

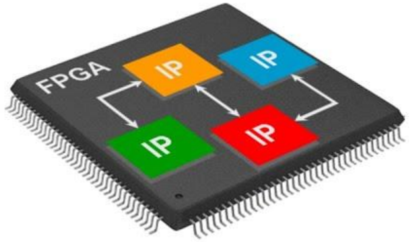
Artwork

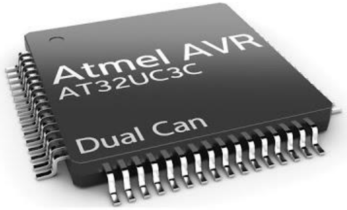


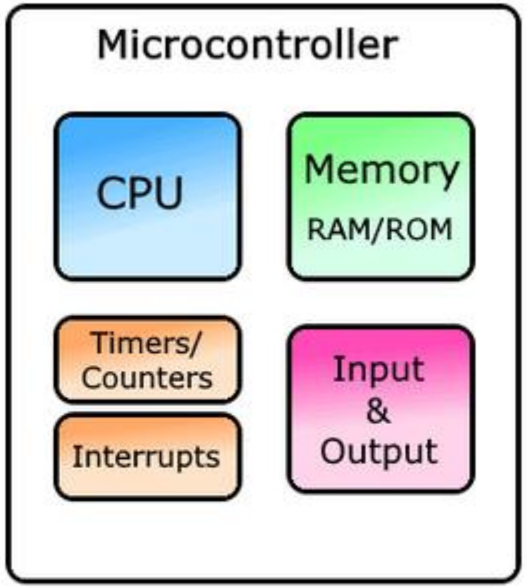


-- Embedded System recap --

- **Embedded System:**
  "Embedded" means hidden inside so one can't see it.
  "System" means multiple components interfaced
  Together for a common purpose.





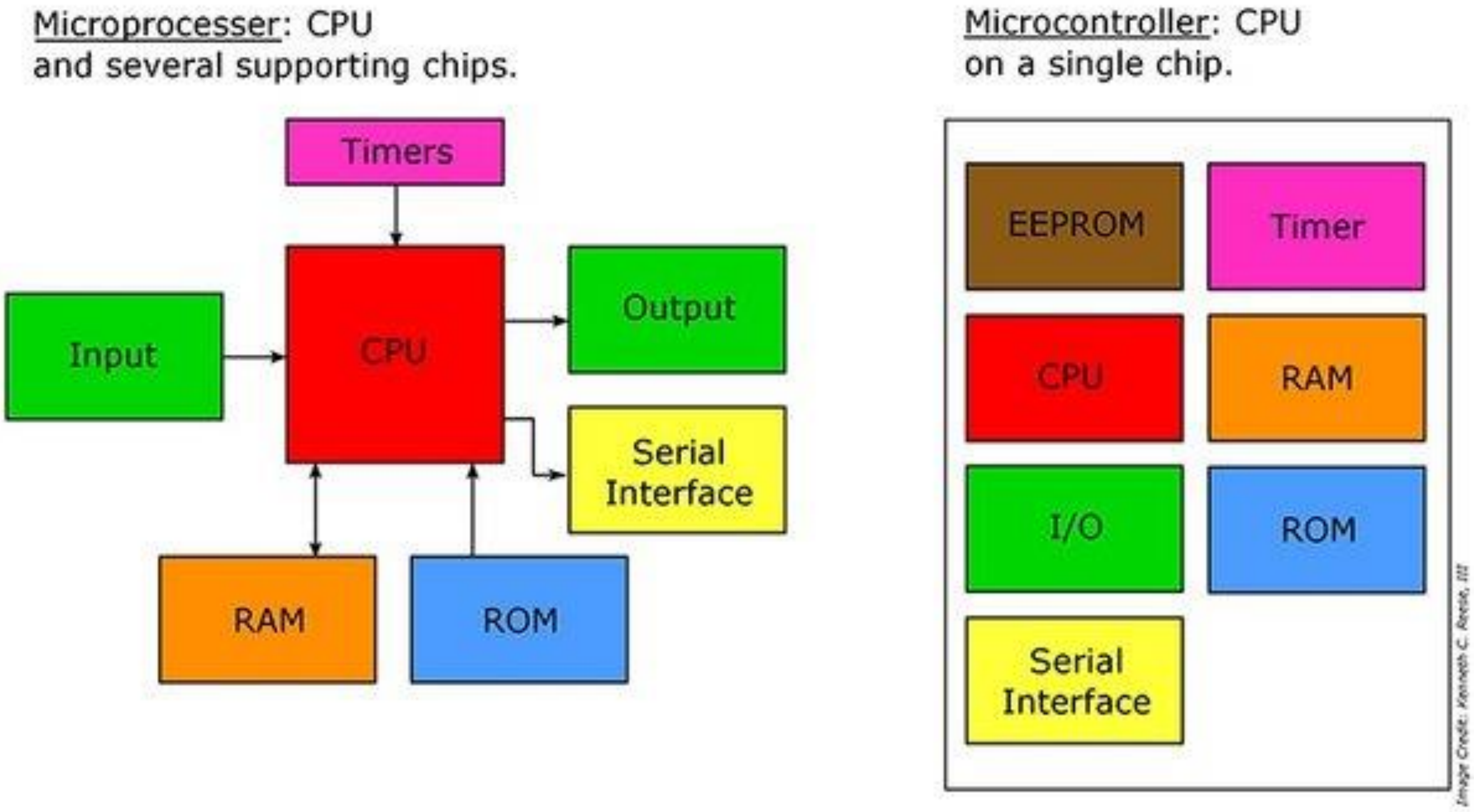


Embedded Memory
Microcontroller

Microcontroller vs Microprocessor

Microprocessor	Microcontroller
Microprocessor is commonly used in general purpose applications like in a computer systems.	Microcontroller is commonly used for application specific purposes like in embedded systems.
Microprocessors widely uses Von Neumann architecture.	Microcontrollers widely uses Harvard architecture.
Microprocessors mainly contain only processing unit, peripherals like RAM, ROM etc. has to be connected externally.	Unlike microprocessors, microcontrollers have peripherals like RAM, ROM etc. embedded into the chip.
The clock speed in which a microprocessor can operate is quite HIGH when compared to a microcontroller. Usually it is above 1GHz.	The clock speed in which a microcontroller can operate is quite LOW when compared to a microprocessor. Usually it ranges from 20MHz to 120MHz.
Microprocessors have very high computational capacity when compared to a microcontroller.	Microcontrollers have less computational capacity when compared to a microprocessor. It is intended for simpler tasks.
Usually microprocessors will have an embedded high performance math co-processor which enables microprocessors to do floating point calculations very fast,	Only high-end microcontrollers will have math co-processors.
Usually circuits associated with microprocessors are complex because of the number of external peripherals attached.	Usually circuits associated with microcontrollers are simple because most of the peripherals are embedded into the chip.
Higher system cost.	Lower system cost.
Higher power consumption.	Lesser power consumption.
Microprocessor based systems are large in size.	Microcontroller based systems are more compact or having small size.
Commonly used in personal computers, laptops, smartphones etc.	Commonly used in washing machines, smart door locks, air conditioners etc.

Microcontroller vs Microprocessor



Harvard Architecture and von Neumann Architecture

Harvard Architecture:

In a Harvard architecture, there are separate memory spaces for program instructions and data. This means that the processor accesses program instructions from one memory (often referred to as the "instruction memory" or "code memory") and data from another memory (typically called "data memory"). Harvard architectures are commonly found in embedded systems and microcontrollers because they can provide separate and dedicated memory access for instructions and data, which can improve performance in certain applications.

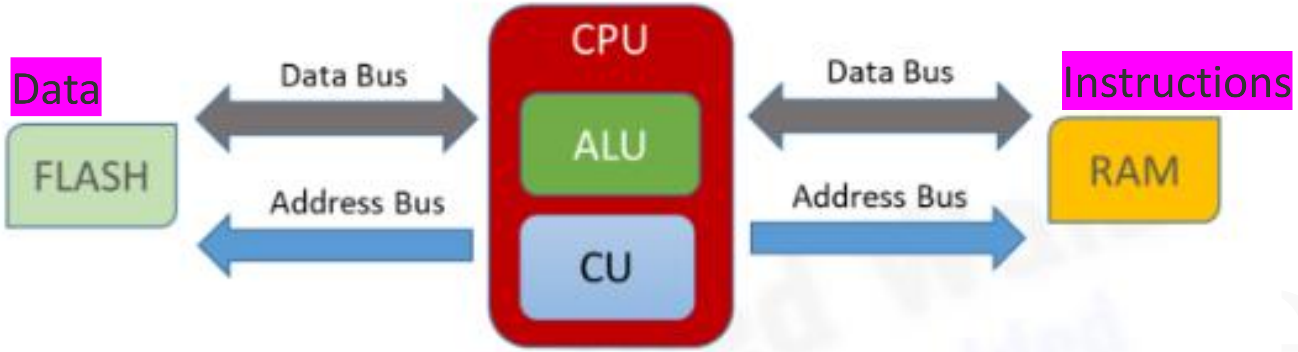
von Neumann Architecture:

In a von Neumann architecture, program instructions and data share the same memory space. The processor fetches both program instructions and data from a single memory unit. This architecture is more common in general-purpose computers, including desktops, laptops, and servers.

Summary of Differences

Feature	Von Neumann Architecture	Harvard Architecture
Memory Structure	Single memory for both instructions and data	Separate memory for instructions and data
Speed and Efficiency	Potential bottleneck due to shared bus	More efficient with parallel access to instructions and data
Flexibility	More flexible, suitable for general-purpose computers	Specialized, used in embedded systems and signal processing

Harvard Architecture and von Neumann Architecture



Harvard Architecture

Von-Neumann Architecture

RISC vs CISC

RISC stands for 'Reduced Instruction Set Computer'.
It is such a design of the CPU that follows simple instructions and is really speedy.

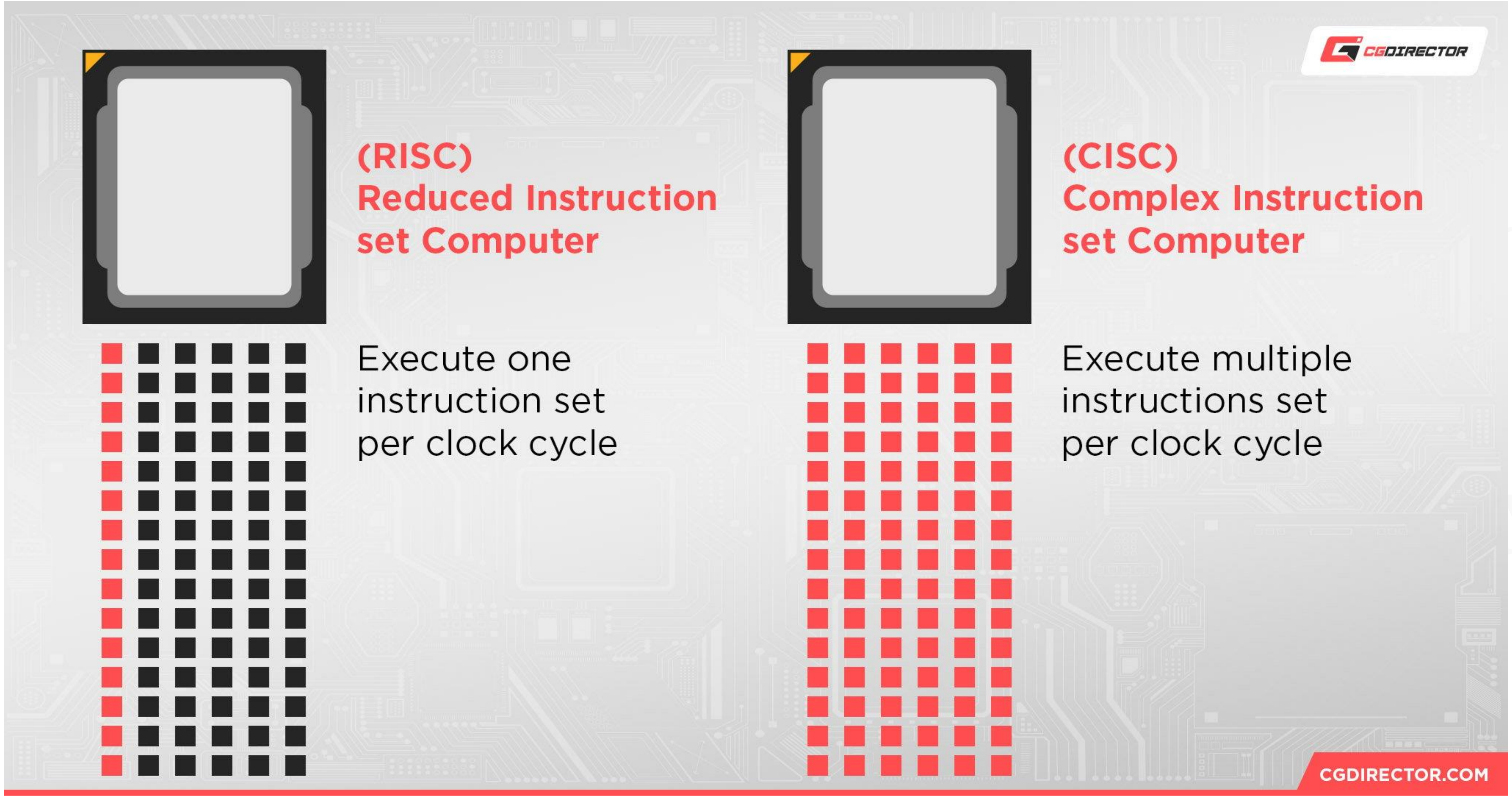
CISC is abbreviated as "Complex Instruction Set Computer".
It is such a design of the CPU that executes a job using only a single command. The command contains multi-step operations that program want to execute. Moreover, CISC machines have relatively smaller programs.

Feature	CISC	RISC
Instruction Set	Large and complex	Small and simple
Number of Instructions	Large	Small
Instruction Execution	Multiple cycles	Single cycle
Instruction Execution Time	Longer	Shorter
Design Complexity	Complex	Simple
Power Consumption	High	Low

RISC vs CISC

CISC	RISC
Emphasis on hardware	Emphasis on software
Includes multi-clock complex instructions	Single-clock, reduced instruction only
Memory-to-memory: "LOAD" and "STORE" incorporated in instructions	Register to register: "LOAD" and "STORE" are independent instructions
Small code sizes, high cycles per second	Low cycles per second, large code sizes
Transistors used for storing complex instructions	Spends more transistors on memory registers

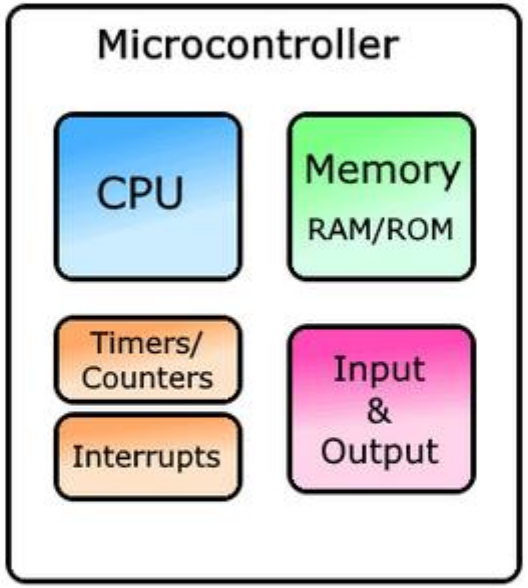
RISC vs CISC


CGDIRECTOR
(RISC)
Reduced Instruction set Computer
Execute one instruction set per clock cycle
(CISC)
Complex Instruction set Computer
Execute multiple instructions set per clock cycle
CGDIRECTOR.COM

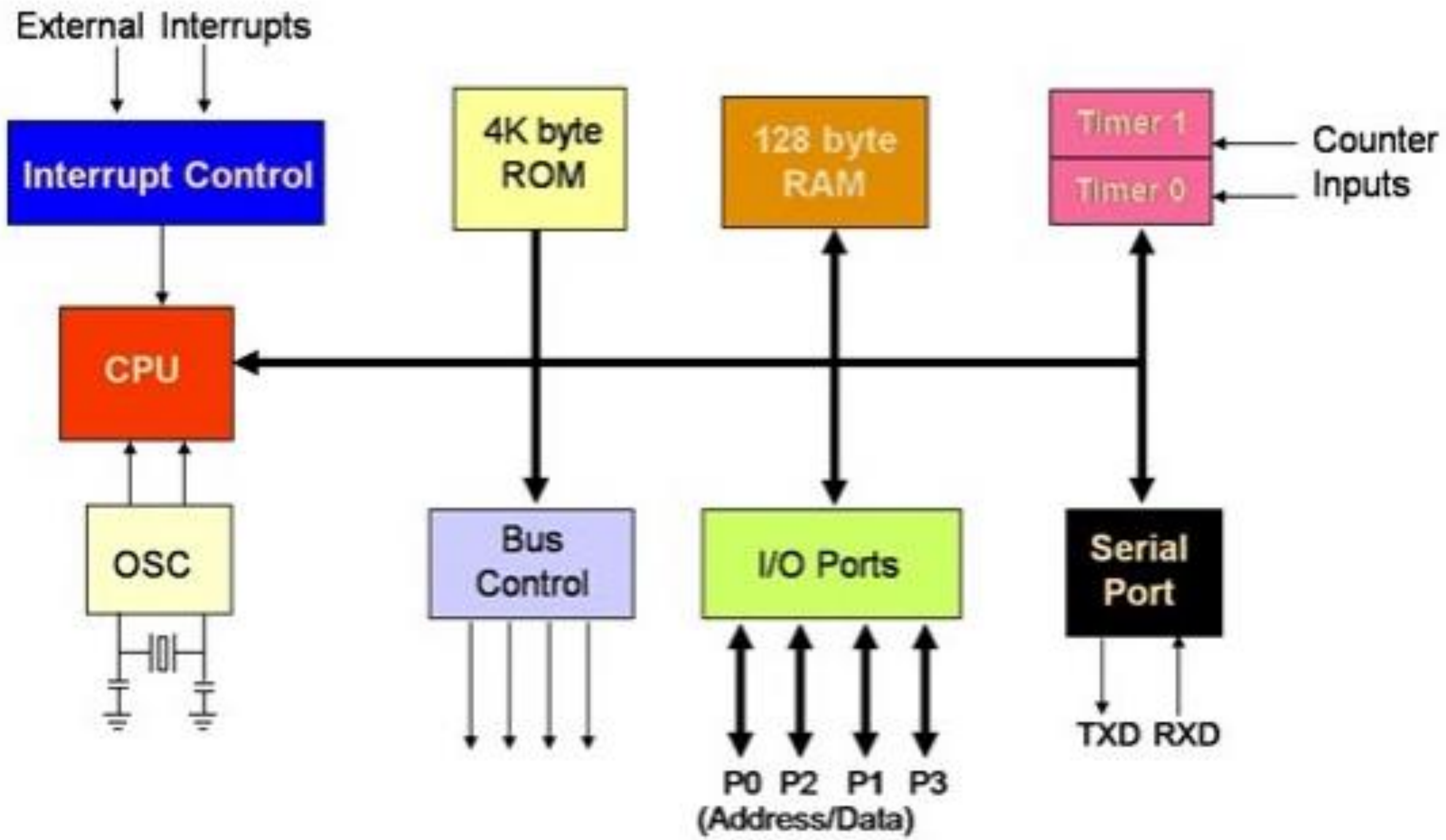
Summary of Differences

Feature	RISC (Reduced Instruction Set Computer)	CISC (Complex Instruction Set Computer)
Instruction Set	Simple, few instructions	Complex, many instructions
Performance	High efficiency, low power consumption	Potentially slower due to more complex instructions
Memory Operations	Load/store architecture (separate memory ops)	Multiple operations in a single instruction
Design Complexity	Simpler CPU design	More complex CPU design
Examples	ARM, MIPS, RISC-V	x86, AMD64 (Intel and AMD processors)

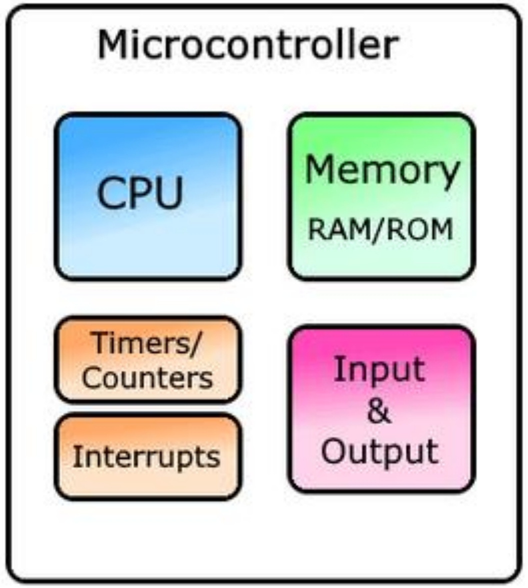
Micro-controller Unit (MCU) Architecture



Embedded Memory
Microcontroller



Micro-controller Unit (MCU) Architecture

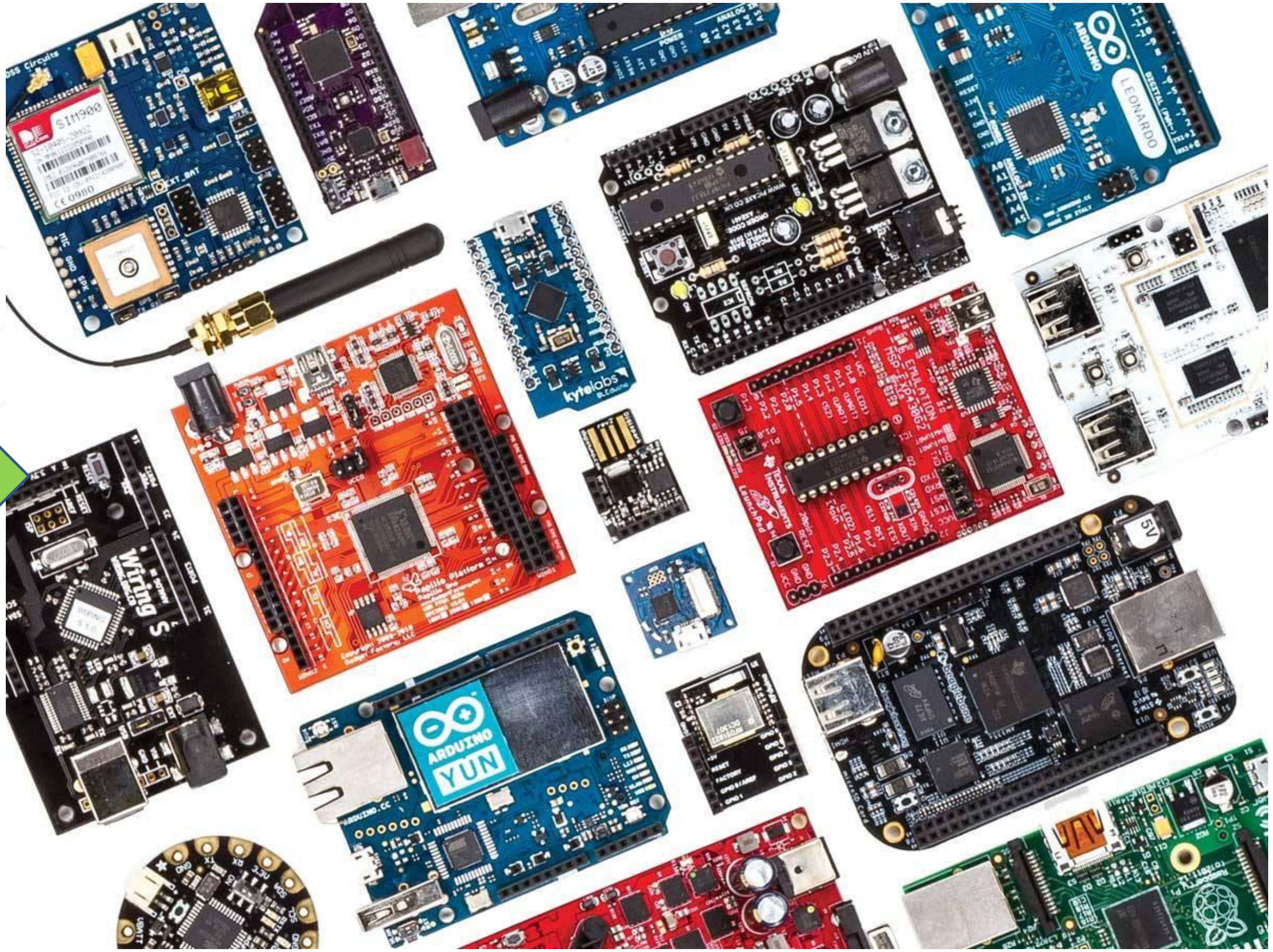


Embedded Memory
Microcontroller

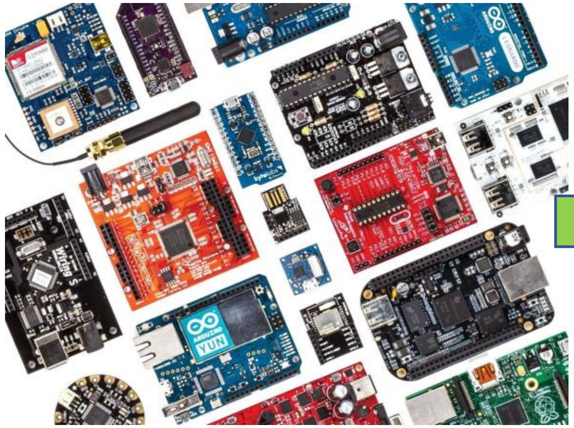
ARM – STM32, NXP, Borsch, TI, ScioSense etc
PICs - Microchip
AVR - Atmel
8051 - Intel

Embedded Systems (Microcontroller) Development Boards

- Arduino
- Raspberry Pi
- Banana Pi
- STM Boards
- ESP32 Boards
- Beagel
- ETC.,

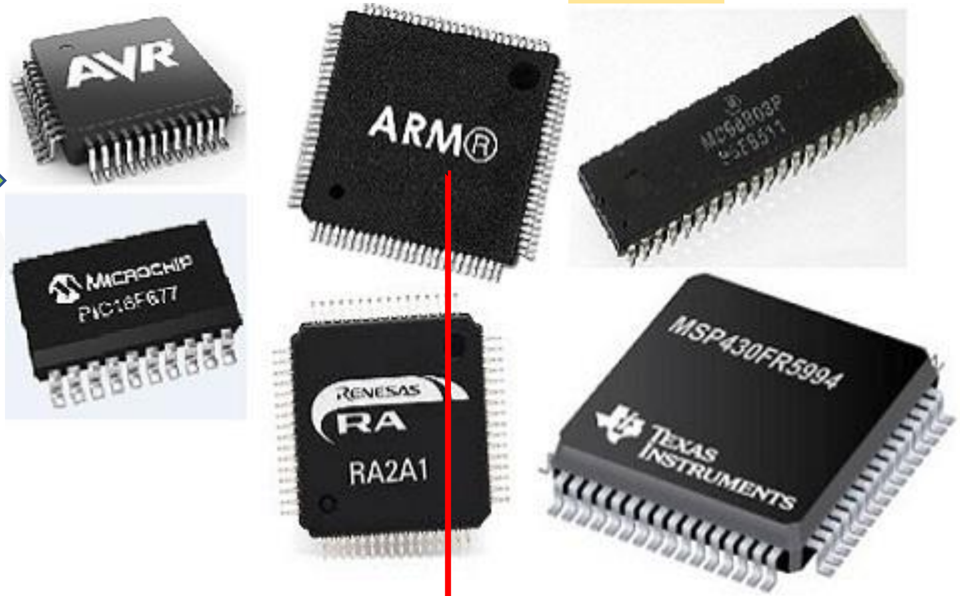


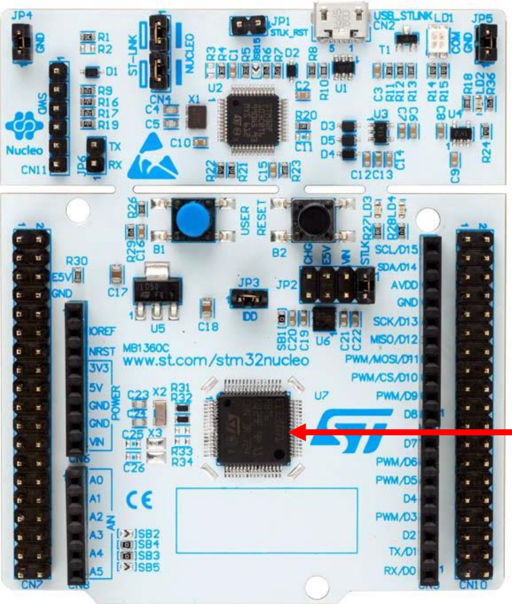
Embedded Systems (Microcontroller) Development Boards



the boards' brains

MCUs





✓ Arduino , most commonly used board for hobbyists

PROS

- Easy to use.
- Great for basic learning experience.
- Affordable.
- Great for hobby projects.
- Straight forward.

CONS

- Limited engineering skills.
- Limited industrial applicability.

✓ STM32 , Industrial level boards

PROS

- STM32 is more practical in the real world.
- Affordable at company work level.
- gets you familiar with ARM Cortex M cores, the most powerful MCU architecture.
- Industrial standard .. Robots, phones etc.,

CONS

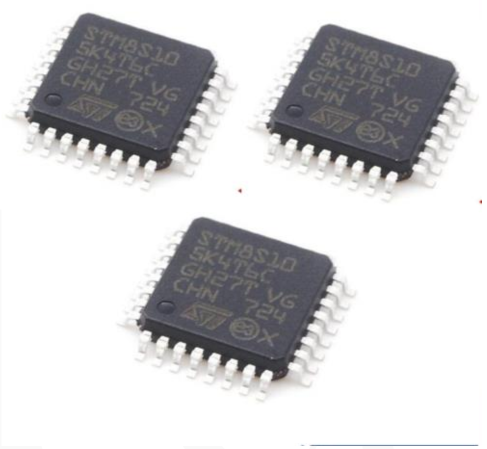
- Needs constant practice.
- Requires practical hardware at all times.

-- Tet la'ngech focuses on STM32 microcontrollers --

	STM32	Arduino
Features	STM32F103	ATMEGA328
Clock Frequency	72 Mhz	16 Mhz
I2C Buses	2	1
SPI Buses	2	1
CAN Bus	Yes	No
Analog Channel	10	8
PWM Channel	15	6
USART Buses	3	1
GPIO's	32	24
On Board RTC	Yes	No
Architecture	ARM Cortex M3 32 bit	AVR RISC 8 bit
ADC Resolution	12 bit	10 bit
Quantization Level	4096	1024
Flash Memory	64KB	32KB
SRAM	20KB	2KB
Debugging	Serial, JTAG	Serial
PWM Resolution	16 bit	10bit
Price	**110**	**115**

-- Tet la’ngech focuses on STM32 microcontrollers --

STM32



*Arm Cortex-M microcontrollers are easily the most popular line of microcontrollers used in commercial electronic products.

*They have been used in tens of billions of devices.

STM32 MCUs are based on ARM Cortex series

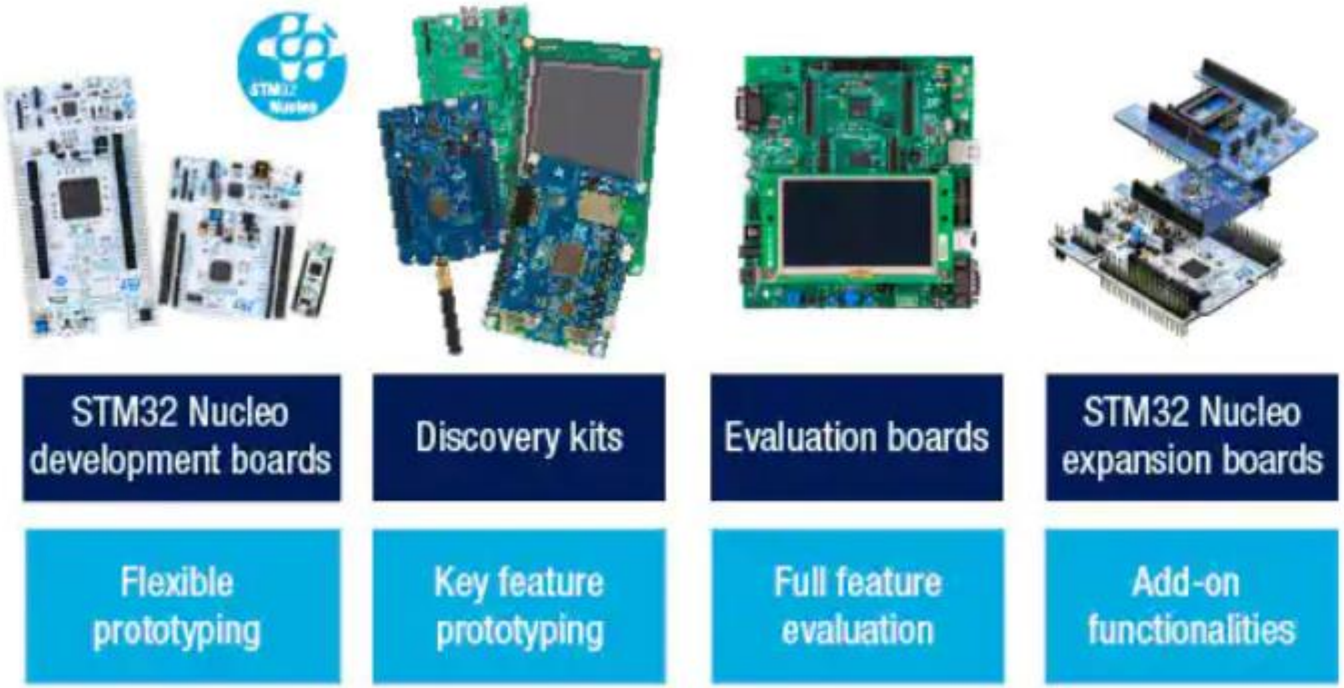
Cortex-A: these processors are built for advanced operating systems and they offer the best performance
Examples; iPhone, Samsung, Huawei, tablets, Smart TVs etc.,

Cortex-R: these processors have fast response time thus they are used for real-time applications.
Examples; Industrial robots, electric cars, airbags etc.,

Cortex-M: Cortex-M processor are specifically built for microcontrollers
Examples; Microwaves, 3D printers, washing machines, Games, Smart speakers etc.,

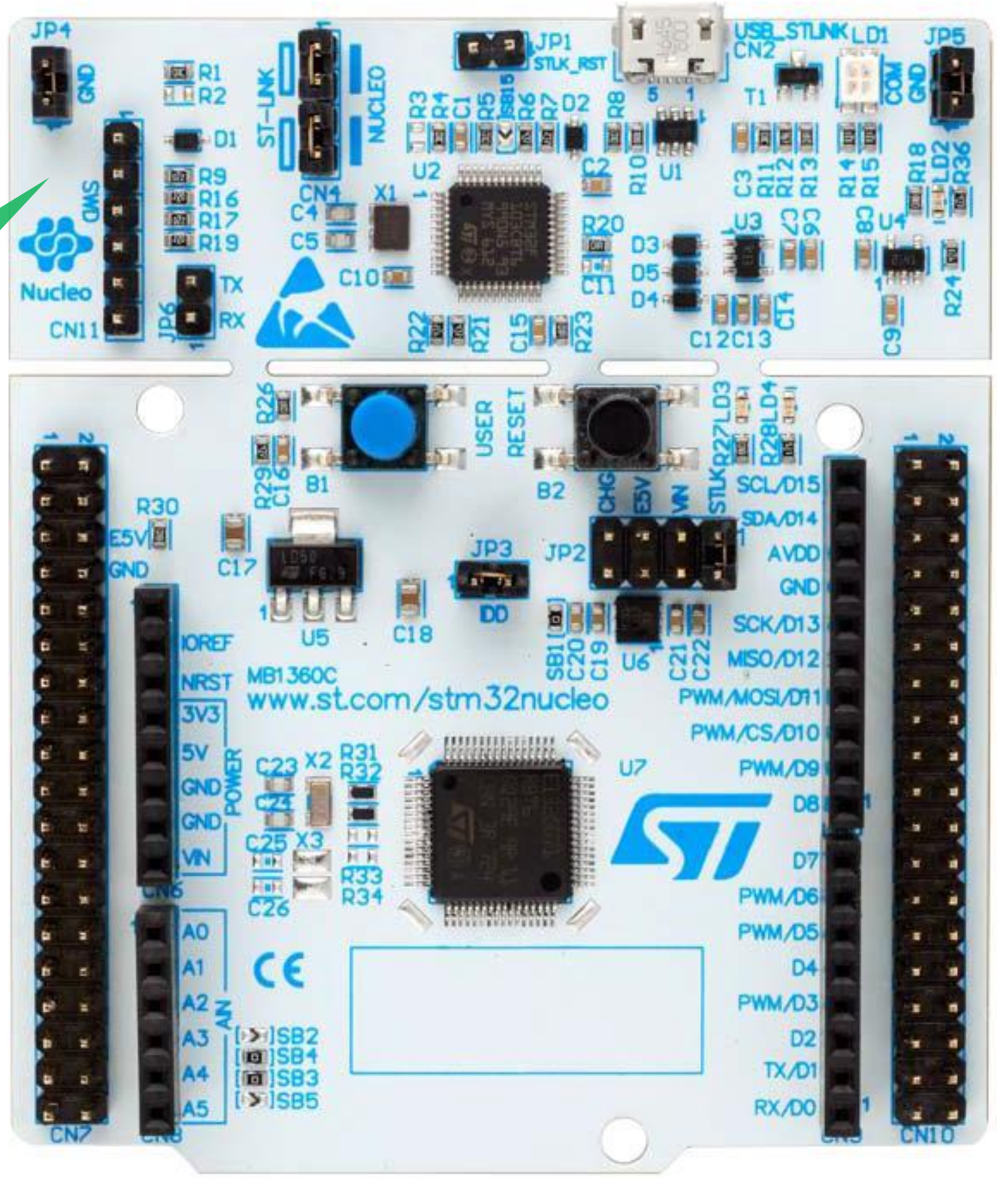
Hardware development boards

- STM Boards



Hardware development boards

We shall be using this STM Board for our practical sessions [STM32nucleo]



STM32 Nucleo Board Layout

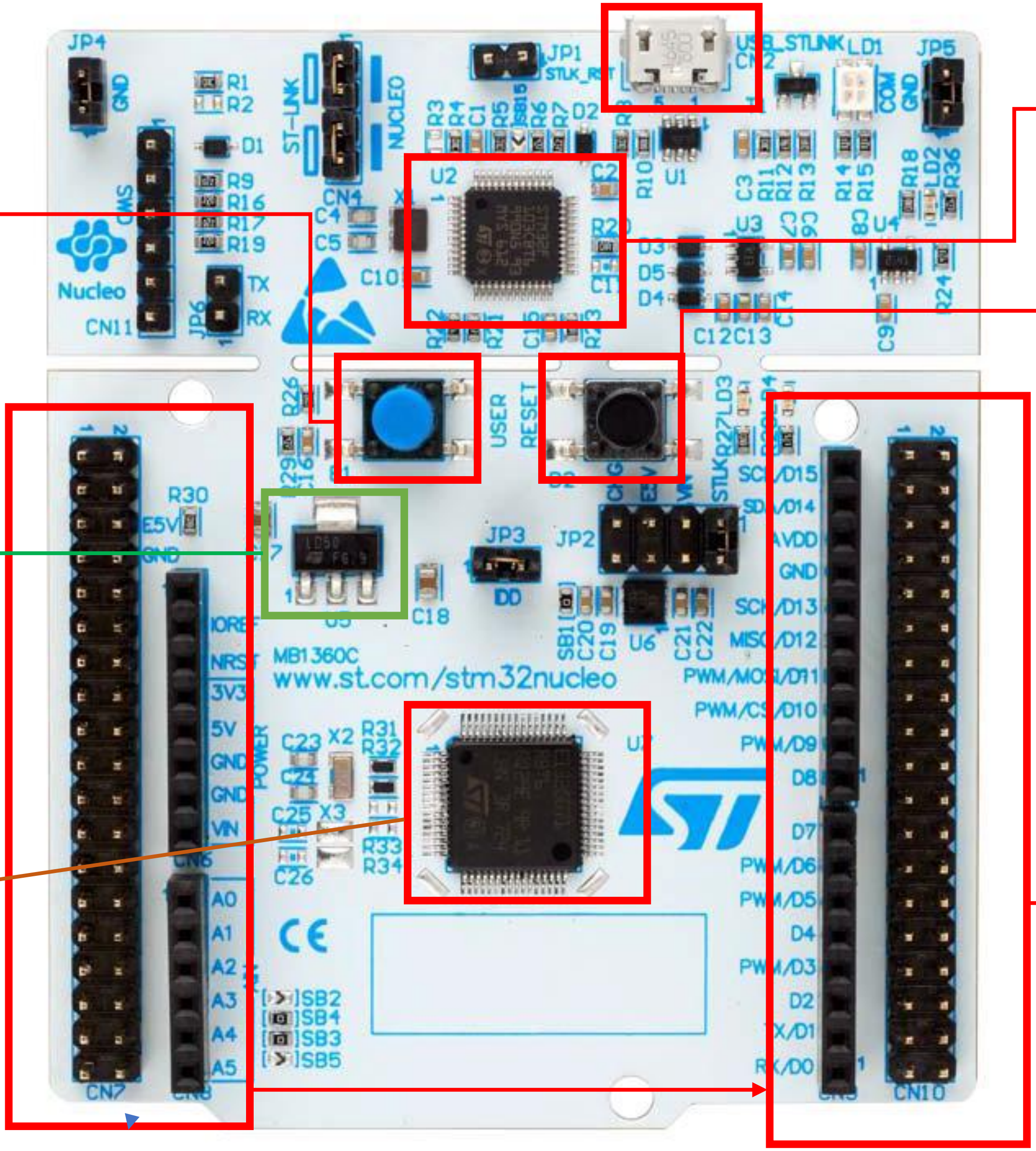
USB-C Connector
Powers the board and used to provide connection between PC and the board

Micro-controller Unit (St-Link)
Controls program download from PC to Main MCU

User Switch
Can be programmed to perform Desired output actions

Reset Switch
Resets main MCU,
re-initiating the running program

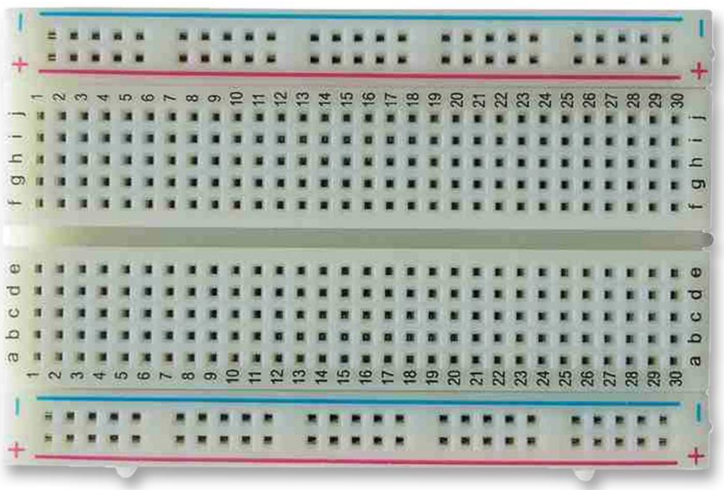
Voltage Regulator
Regulates and distributes desired power to the components

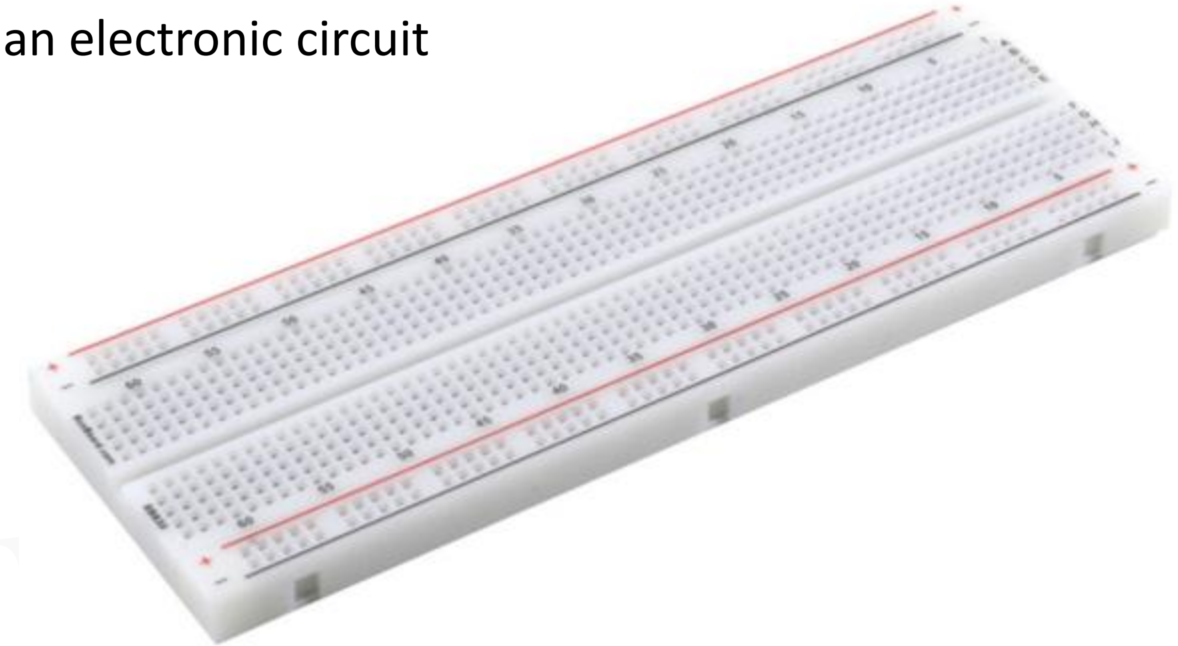


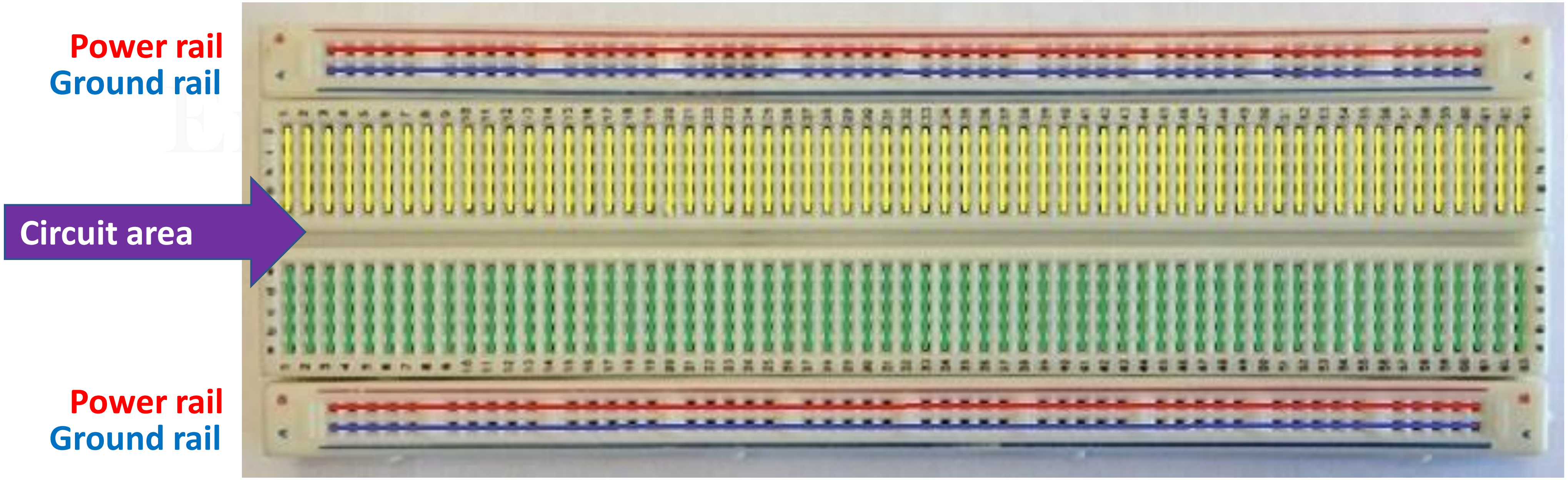
Micro-controller Unit MCU (Main)
Brains that control the board components

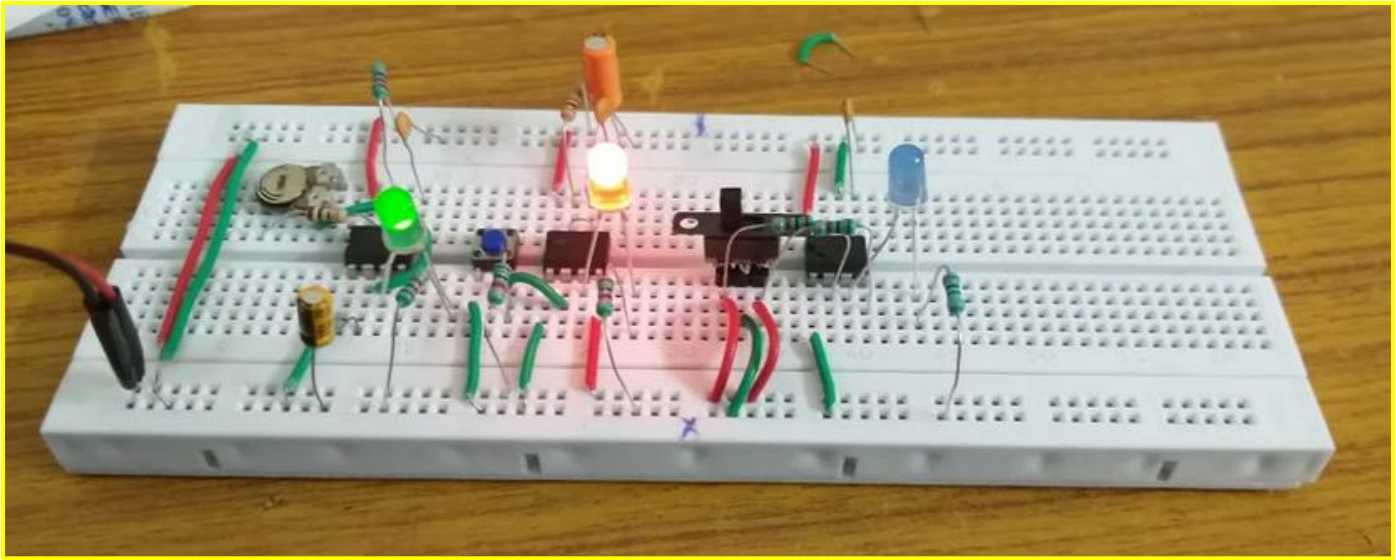
External Connectors
Connection to Inputs & Outputs
- Sensors
- Displays
- LEDs
- Speakers
- etc.

Breadboard - is a non-solderable board used to set up an electronic circuit

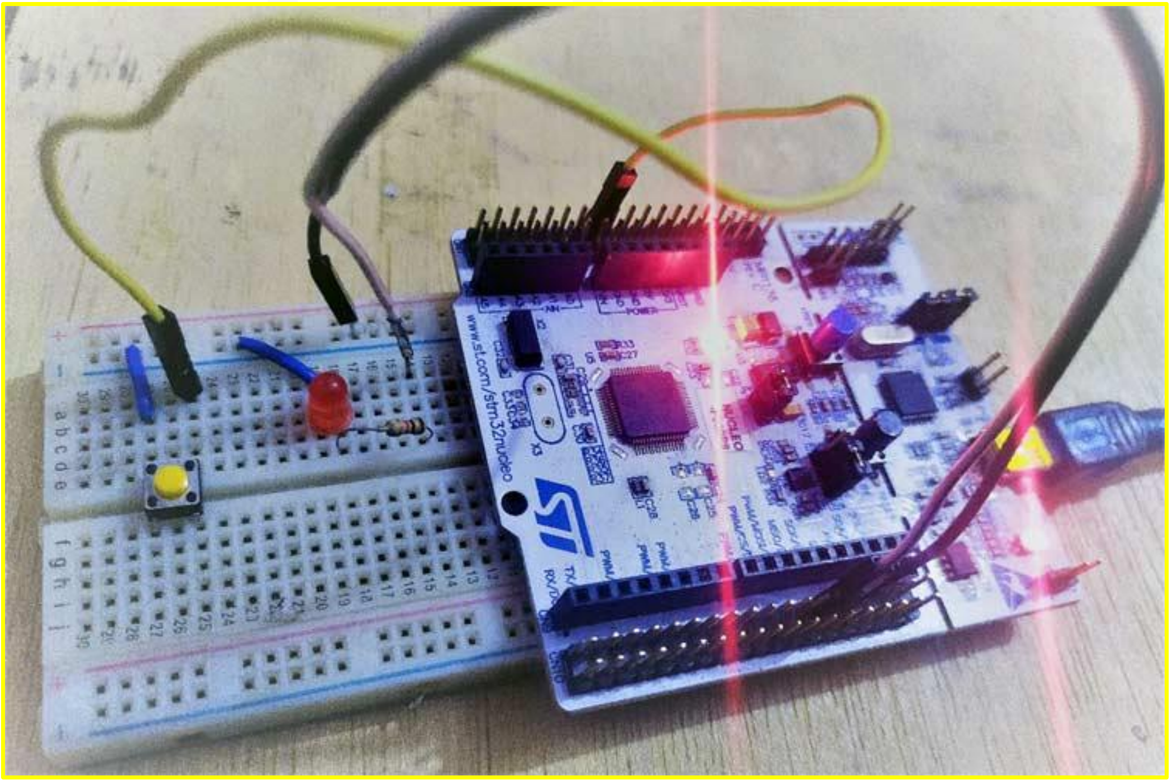




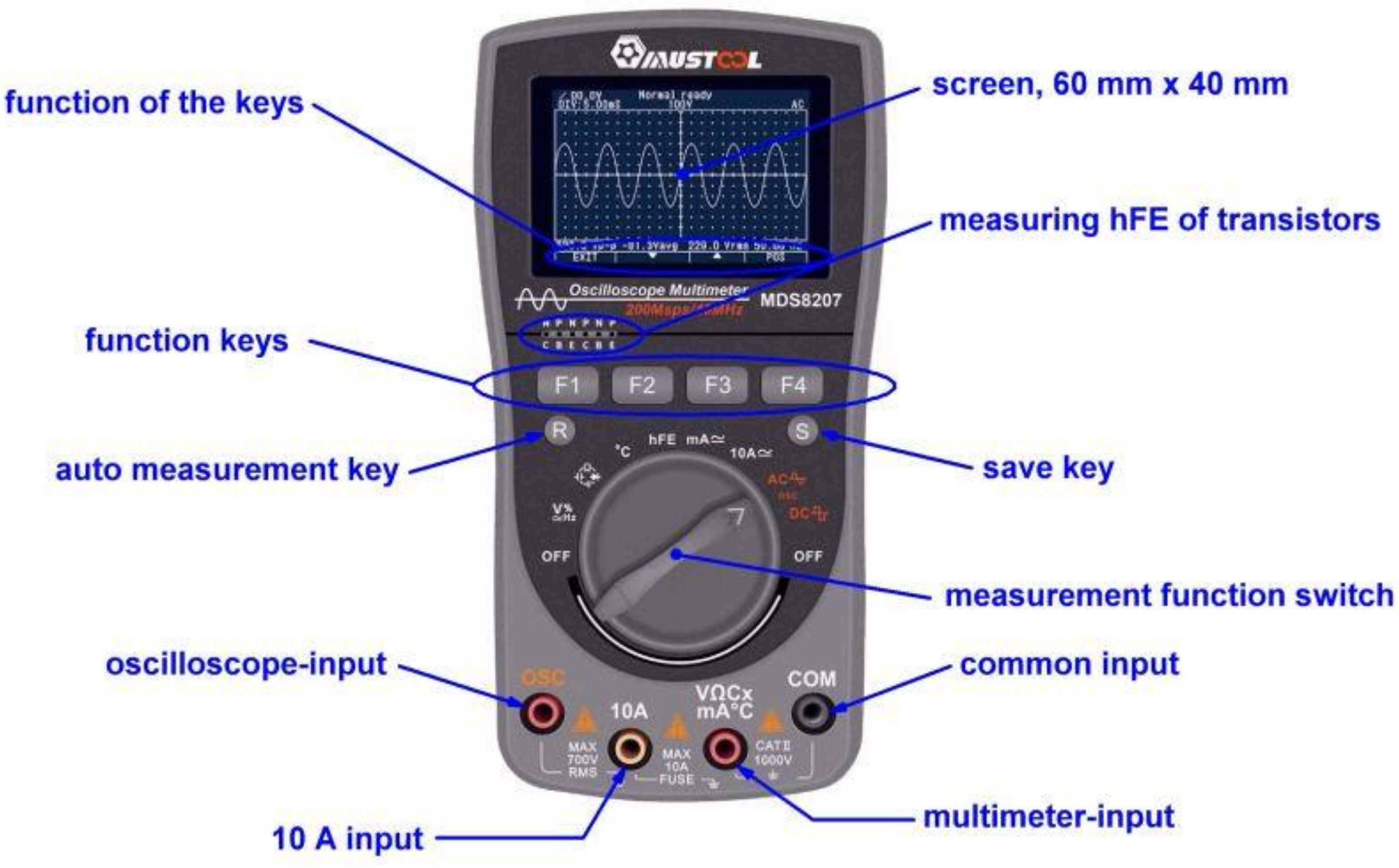




STM Board Connection
to Breadboard to
make a circuit project



Basic Tools - Multimeter



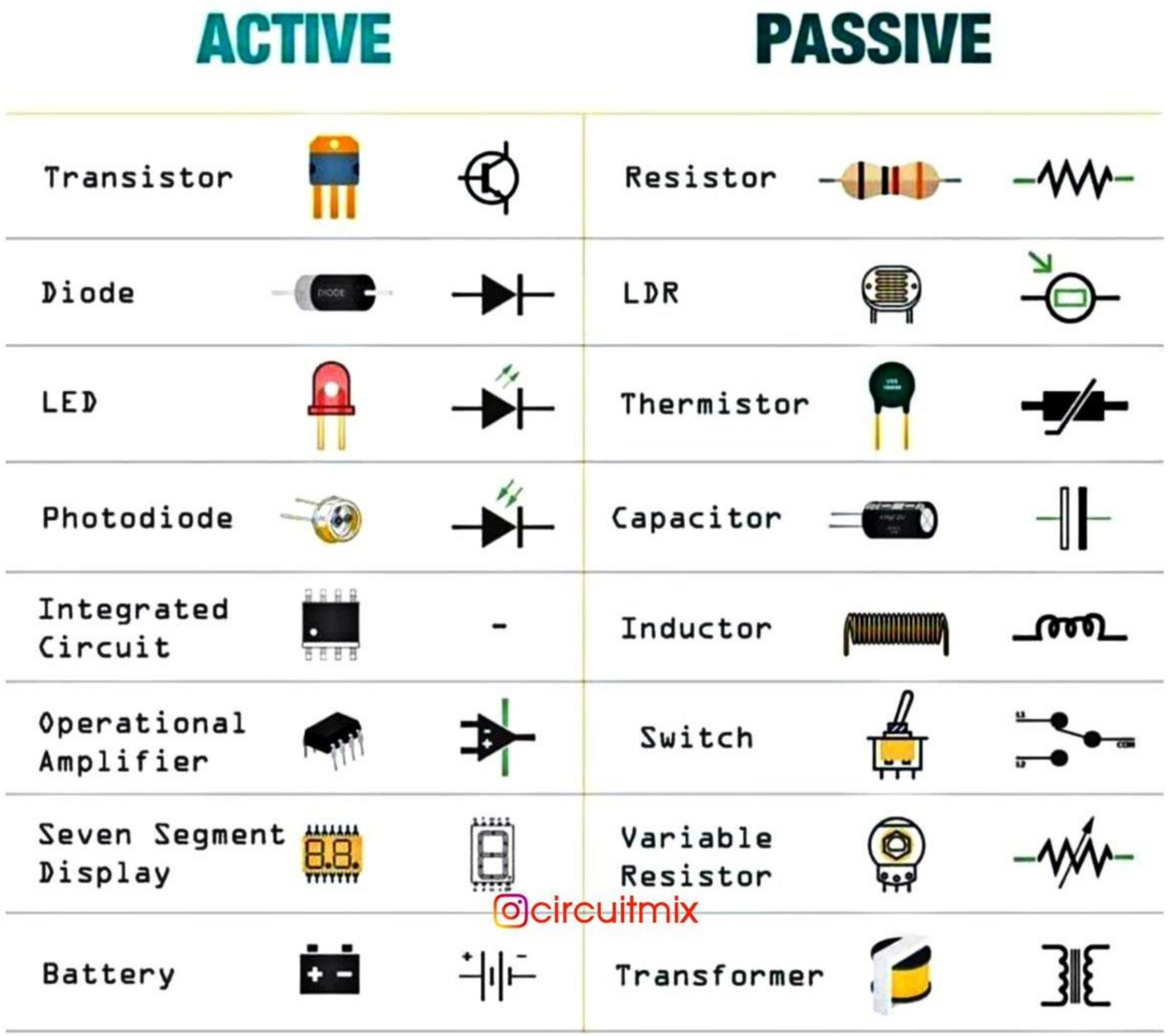
Measures;

1. Current
2. Voltage
3. Frequency
4. Temperature
5. Capacitance
6. Digital sine wave

Basics for a hardware engineer

1. Resistance
2. Capacitance
3. Current
4. Voltage
5. Frequency
6. Ohm's law
7. Inductance

How
i) Diodes work
ii) Capacitors work
iii) Resistors work
iv) Inductors work
v) Current and voltage flow



Basics for a hardware engineer

Sensors

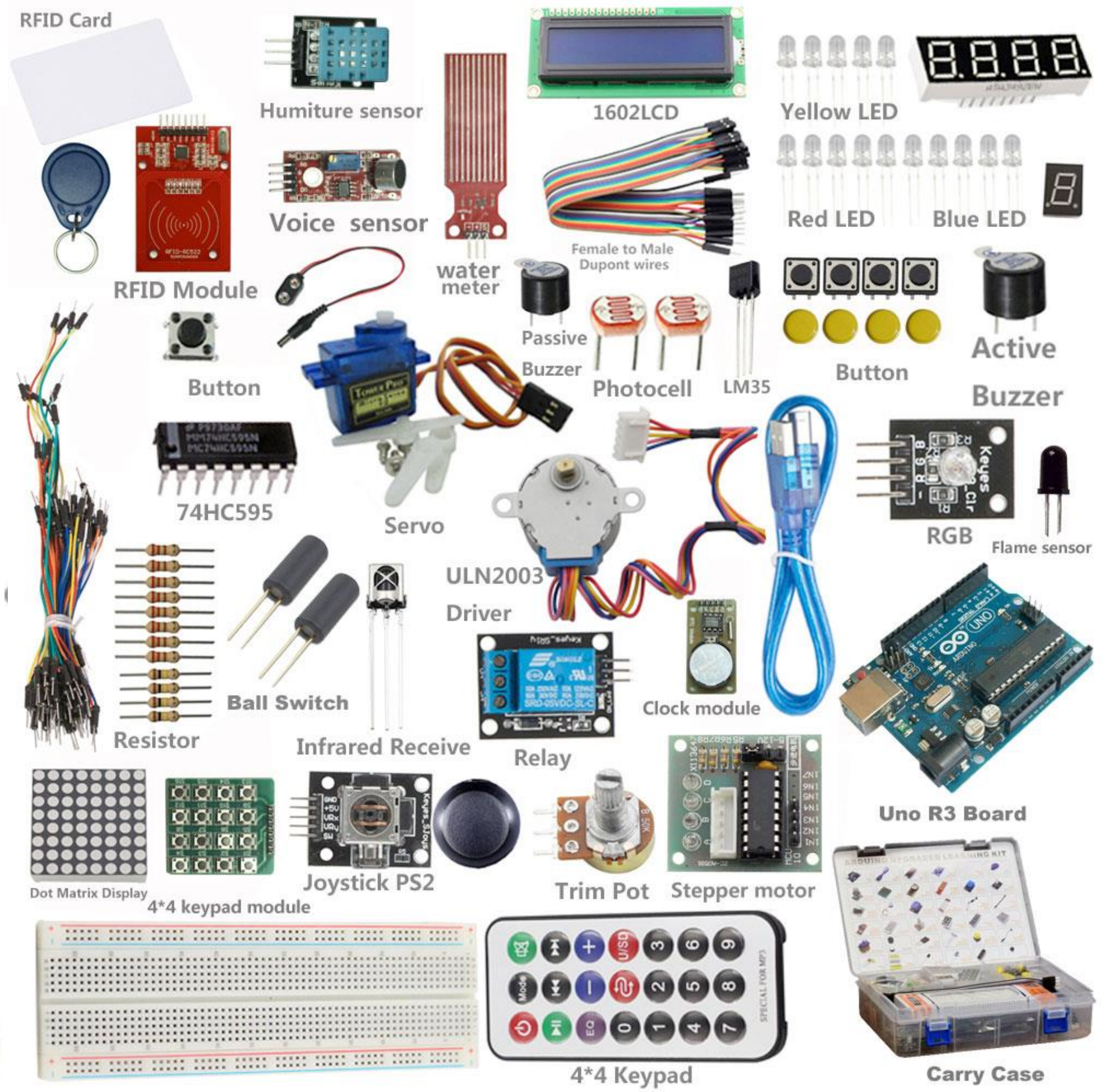
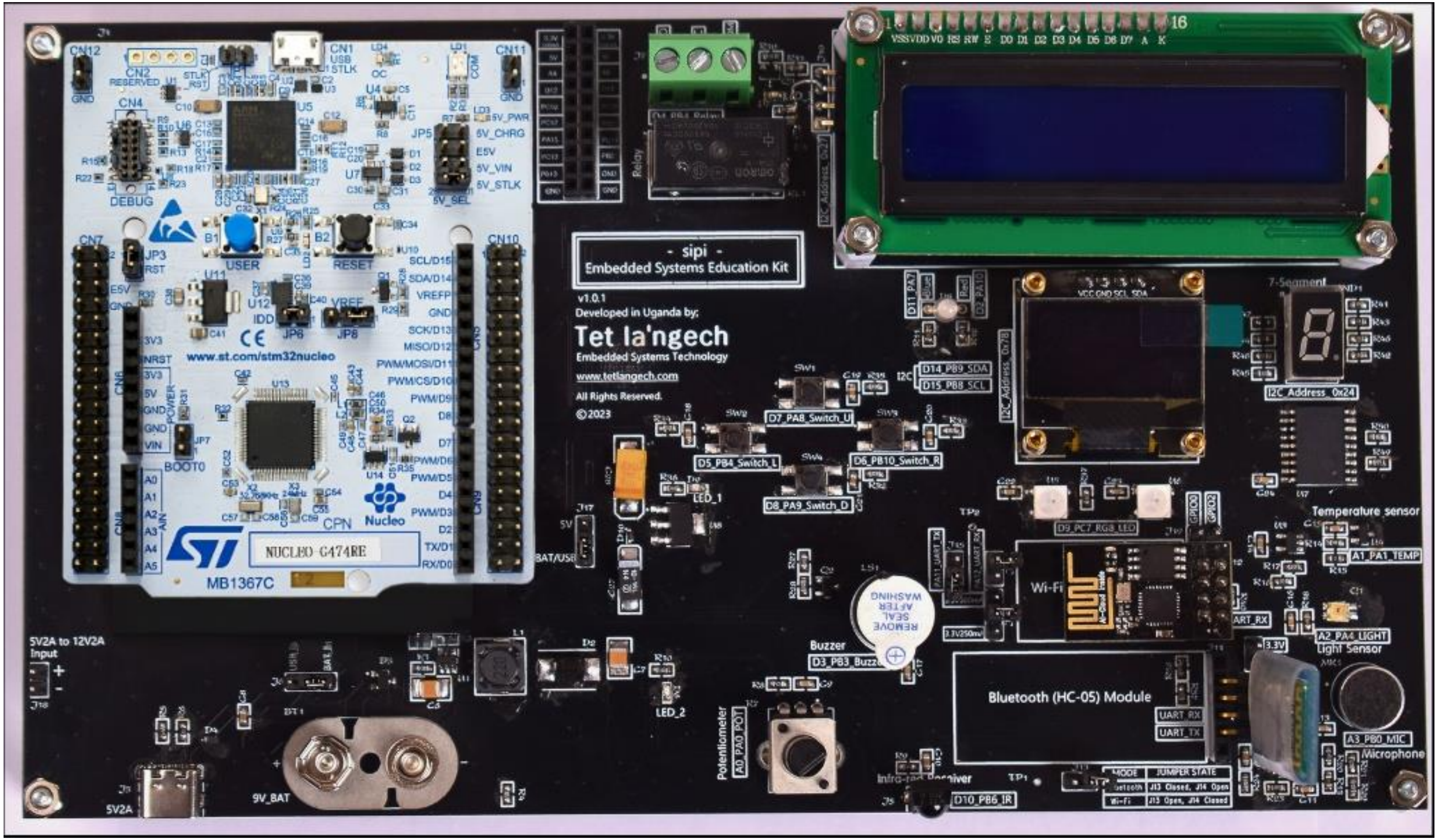


Image rights: surplustronics

sipi education kit from Tet la’ngech Ltd.,



Designing a Custom PCB Board

Steps:

1. Project Plan (involves Hardware, Firmware and Product Design Engineers)
2. Block Diagram design
3. Major Components search
4. Schematics Development and Design
5. Artwork Design
6. Gerber files generation

7. PCB Fabrication (Third party fabricators)

8. Component placement and soldering (can be done manually or by Auto Placement Equipment)
9. PCB Testing and Debugging (order: Connectivity > power > others)
10. Ready for Firmware upload
11. Testing continues together with Firmware Engineer
12. Board is passed on to product designer for casing.
13. Final Functionality testing by all the 3 teams

Designing a Custom PCB Board

Schematics, Artwork Design & Development:

1. orCAD
2. Altium
3. PADs
4. KiCAD
5. Proteus
6. Easy EDA
7. Eagle
8. DipTrace
9. etc.,

** A hardware Engineer is the most important in Embedded Systems Development,

there's barely no room of mistakes that will lead to system failure.

“Transforming Uganda through advanced embedded systems technology”

Get In Touch **With Us**

Contact **Us**

For more information and inquiries

PB107, Mechatronics Department
Uganda Industrial Research Institute,
Namanve, Mukono Uganda

+256 700 567 217 +256 761 720 105

tetlangech@gmail.com

www.tetlangech.com

@tetlangech