Getting Started with KuttyPy

Jithin B.P

February 22, 2019

Download necessary files

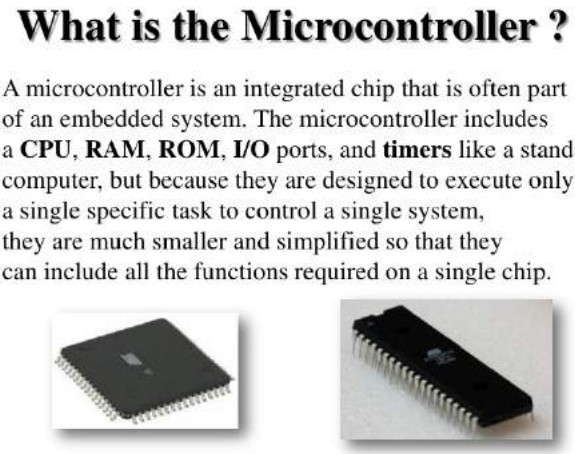
Network

connect to wifi network 'homepc' , password 'industry4.0'

Download Zip

- open a web browser
- 10.42.0.1:4000/kuttypy.zip

Microcontrollers



What is the Microcontroller ?

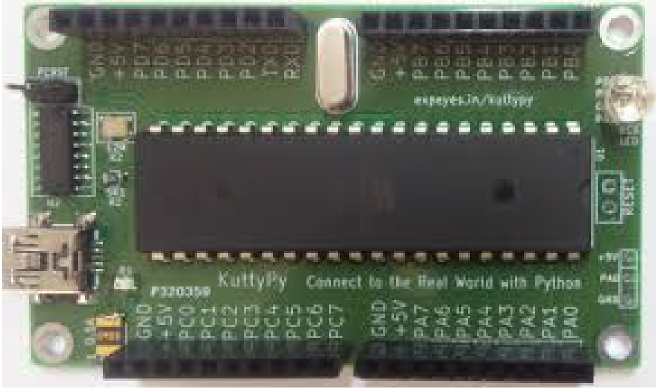
A microcontroller is an integrated chip that is often part of an embedded system. The microcontroller includes a **CPU**, **RAM**, **ROM**, **I/O** ports, and **timers** like a standard computer, but because they are designed to execute only a single specific task to control a single system, they are much smaller and simplified so that they can include all the functions required on a single chip.

- Found in almost every appliance/gadget. E.g. microwaves, music players, automatic doors, elevators, cars, lab instruments etc
- Designed to efficiently handle simple tasks such as monitoring a switch and taking appropriate action after checking for other parameters.

What is a development board?

A development board is a circuit board which has minimal external components required by a microcontroller such as

- Connectors for easily accessing the pins
- power supply socket and voltage regulator
- A crystal oscillator which decides the speed of operation of the program.

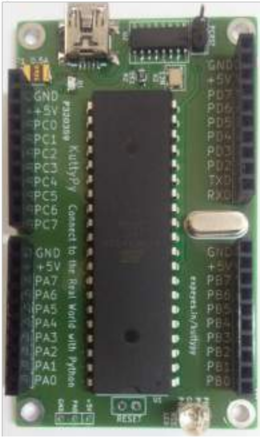


What is KuttyPy?

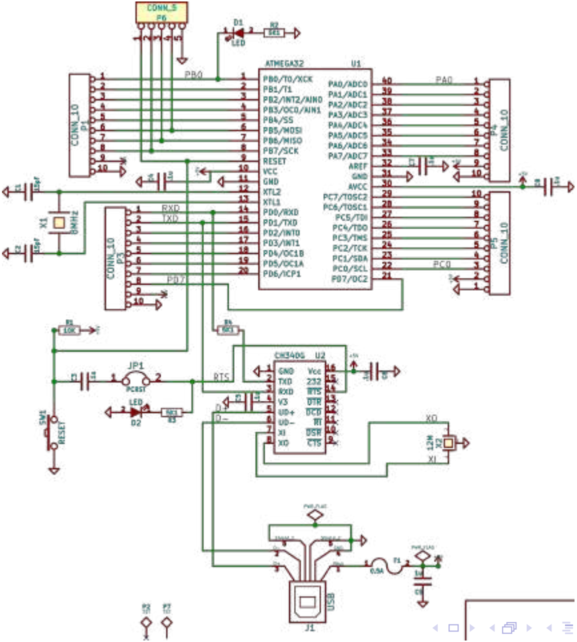
KuttyPy

INTERACTIVE PLAYGROUND
[MICROCONTROLLER TRAINING UTILITY]

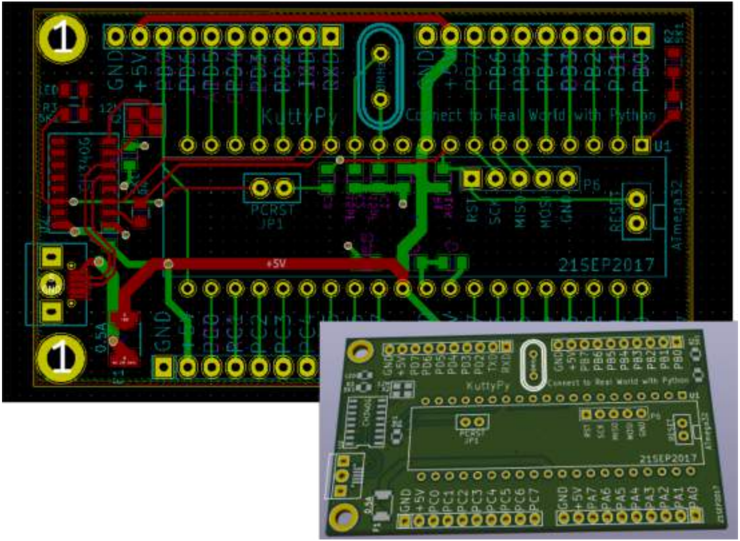
* Software-RealWorld Bridge
* Bootloader with built-in register R/W access
* Functions as a regular Atmega32 microcontroller development board
* A tool for learning microcontrollers



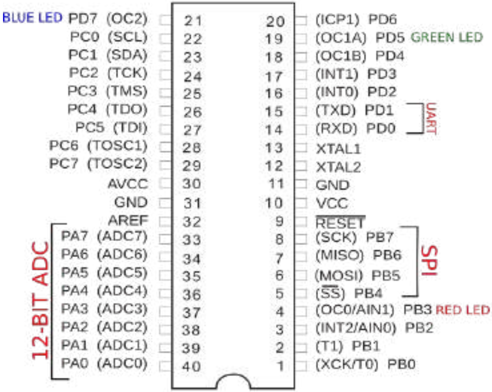
Schematic

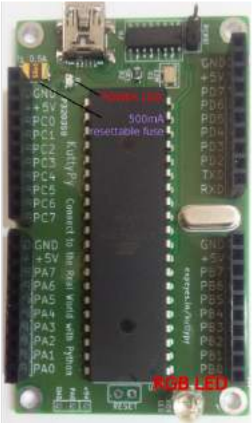

CONN_5
P6
D1
LED
R2
ATMEGA32
U1
PB0
PB0/T0/XCK
PB1/T1
PB2/INT2/AIN0
PB3/OC0/AIN1
PB4/SS
PB5/MOSI
PB6/MISO
PB7/SCK
RESET
VCC
GND
XTL2
XTL1
PD0/RXD
PD1/TXD
PD2/INT0
PD3/INT1
PD4/OC1B
PD5/OC1A
PD6/ICP1
PA0/ADC0
PA1/ADC1
PA2/ADC2
PA3/ADC3
PA4/ADC4
PA5/ADC5
PA6/ADC6
PA7/ADC7
AREF
GND
AVCC
PC7/TOSC2
PC6/TOSC1
PC5/TDI
PC4/TDO
PC3/TMS
PC2/TCK
PC1/SDA
PC0/SCL
PD7/OC2
PA0
PC0
CONN_10
P1
CONN_10
P3
CONN_10
P4
CONN_10
P5
RXD
TXD
PD7
X1
8MHz
15pF
15pF
C1
C2
R1
10K
R4
JP1
PCRST
RTS
LED
D2
R3
SW1
RESET
CH340G
U2
GND
TXD
RXD
V3
UD+
UD-
XI
XO
Vcc
232
RTS
DTR
DCD
RI
DSR
CTS
D+
D-
XO
XI
12M
X2
PWR_FLAG
PWR_FLAG
F1
0.5A
C6
USB
J1
P2
P7

Layout


GND
+5V
PD7
PD6
PD5
PD4
PD3
PD2
TXD
RXD
KuttyPy
Connect to Real World with Python
GND
+5V
PB7
PB6
PB5
PB4
PB3
PB2
PB1
PB0
R2
5K1
LED
R1
5K1
CH340G
U1
P6
PCRST
JP1
RST
SCK
MISO
MOSI
GND
RESET
ATmega32
21SEP2017
+5V
0.5A
GND
+5V
PC0
PC1
PC2
PC3
PC4
PC5
PC6
PC7
GND
+5V
PA7
PA6
PA5
PA4
PA3
PA2
PA1
PA0

I/O pins: organisation

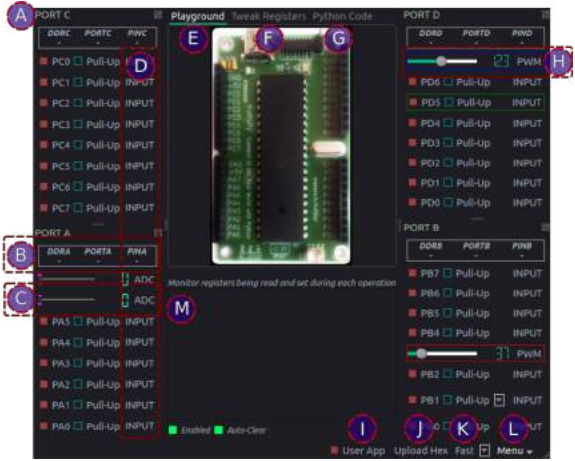




All pins are grouped into their respective ports

Each port is a 10 pin berg socket with 8 I/O pins, 5V, and Ground.

Software



Python based control and commuication library, and GUI

All register manipulations carried out via the GUI are clearly displayed in the log window.

Differences with Arduino: Bootloader specifics

BOOTLOADER

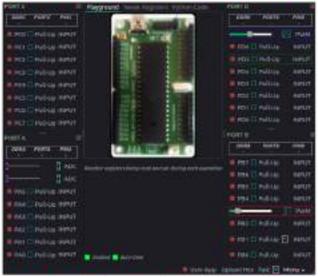
+ Read hex files via the serial port, and write to flash.

+ Start executing user code if no serial data is received during boot

User code area: ~30kB free

ARDUINO

FLASH MEMORY



Example: Photo of a persistence of vision display with 8 LEDs on PORTC

BOOTLOADER

+ Interpret serial commands for reading and writing registers associated with the microcontroller (setReg, getReg). Study registers without the compile-upload hassle.

+ User the Python library to make kuttyPy a software - real_world bridge.

+ Read hex files via the serial port, and write to flash.

+ Start executing user code if no serial data is received during boot

User code area: ~30kB free

KUTTYPY

FLASH MEMORY

Differences with Arduino: Learning to use microcontrollers

Arduino

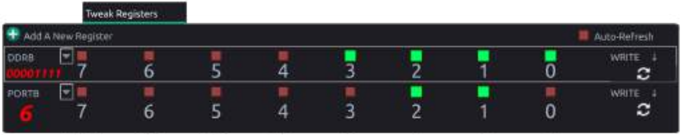
- Code must be compiled and uploaded to test behaviour
- PORTS and bit manipulations are replaced with pin numbers annd high level functions, thereby hiding the microcontroller architecture.

KuttyPy

- Real-time manipulation and readback of registers via the serial communication port.
- Graphical utility for quickly checking behaviour.
- Controlled via Python running on a traditional PC. Python modules can be used to develop complex projects.
- Code can also be compiled and uploaded for standalone operation.
- PORTS are classified as is, and students are encouraged to use bit manipulation and understand the relevance of binary.

The importance of binary

Microcontroller programming revolves around binary, and registers (variables whose bits have specific hardware duties)



DDRB 00001111

PORTB 6

128 64 32 16 8 4 2 1

Binary

00001111 = 0 + 0 + 0 + 0 + 8 + 4 + 2 + 1 = 15

DDRx : Each bit decides if the corresponding pin is input(0) , or output (1)
DDRB = 15 = 00001111 => PB7, PB6, PB5, PB4 are inputs.
PB3, PB2, PB1, PB0 are outputs

PORTx : Each bit decides if the corresponding pin is connected to 5V(1) , or Ground(0) .
DDRB = 6 = 00001100 => PB7, PB6, PB5, PB4 are inputs.
PB3 , PB0 are at 0 Volts
PB2, PB1 are at 5Volts

Project ideas using kuttyPy with Python

combined with various visualization and analytical modules of Python, several applications can be thought of.

- OpenCV : This image processing tool can be used to interpret webcam data, and create a motion tracking tool with KuttyPy. Moving camera mounts can be made using stepper motors controlled by KuttyPy.
- Matplotlib: This simple plotting tool can be used to create a voltage data logging tool with a few lines of code. The ADC of the kuttypy reads voltages in a 0-5Volts range, and returns a proportional number in the 0 - 1023 range.
- pymouse : Use an analog joystick connected to the kuttypy to move the mouse cursor. This can be used to develop assisted input technologies.
- Healthcare : Pulse monitoring, and automated, continuous analysis.