<http://mcdude.github.io/minicore/package_MCdude_MiniCore_index.json>

for the flight controller we need to know the number of output.

Linear acceleration: - accelerometer

Angular acceleration: - gyro meter

* Accelero-gyro-board: -MPU6050
* There 7 axes for a drone.
* we measure the altitude in the quadcopter by barometer.
* For a GPS used drone we need a magnetometer to sense the direction.
* ATMEGA 328p
* R8 -> protects the LED from back EMF
* Out of these 32 pins we have pin(0) to pin (13) are there to control the device
* UARD pins- universal Arduino receiver pins -> these pins are used to connect the
* The hard ware are connected through the 0 and 1 pins
* 2-13 pins are digital IO PINS
* 3,5,6,9,10,11 are used additional feature to PWF.
* 10,11,12,13->Ss misomosi sck-> ICSP in circuit system programmable
* A0- A7 works like analogue input
* A0-A5 additional feature of digital io (input output pin)
* For front left pin(12)
* For front right pin(11)
* For back right pin(7)
* For back right pin(8)
* For r LED pin a1, g LED pin a0, b led a2.
* For the motor the pins are 5,6,9,10.
* If the program wont run just reverse the wire.
* Red chip -> ground, tx, rx,
* Rotation of the LED with r,g,b clock wise and anticlock-wise

**SASWAT DAS NOTE**

IIIT BBS – Robotics Workshop

**DAY 1**

-        Arduino IDE

-        Modded chipset provided; not an Arduino per se

-        Install bootloading software called MiniCore

-        PWM – Pulse Width Modulation (important) – Responsible for controlling the speed of the motor and control the amount of voltage supplied to the motors

-        TTL -  Measure of voltage supply – most standard devices require only 5 volts to function

-         https://mcudude.github.io/MiniCore/package\_MCUdude\_MiniCore\_index.json

Downloads (dynamically) the MiniCore software into the module and installs board software onto the IDE

-        Installed MiniCore – Board Module

-        Required hardware – Chipset, Battery, Wifi enabled controller, USB signal conveyer to board

-        Building a Quadcopter

-        Check for the no. of outputs on a flight controller, which depends on the no. of motors the drone has.

-        Quadcopter – Three rotors, one servo

-        Sensors – Accelerometer (Measures tilting by comparing the orientation with the gravitational axis and measures linear acceleration, unaffected by rotation), Gyrometer (Measures angular acceleration, detects rotation, affected by rotation)

-        Chip name – MPU6050 – Accelero-Gyro Board

-        6 translational axes

-        Sensor that detects altitude – (Ultrasonic may help but is useless after a certain range) Altimeter fitted with a barometer; altitude based stabilisation. (7th axis ; Altitude hold)

-        When the quadcopter moves forward or turns, it experiences height damping which may reduce its altitude for which we use the barometer

-        8th Axes – The Earth’s magnetic field – Sensed by Magnetometer

-        Indications – LEDs used for this purpose to communicate system status to the user or a person performing error diagnosis on it.

-        SMD components used to make the circuit board compact

-        ATMEGA 328P – Chip being used – 2 KB RAM, 32 KB ROM

-        Bootloader for MiniCore already available in the chipset

-        Drivers required to run motors

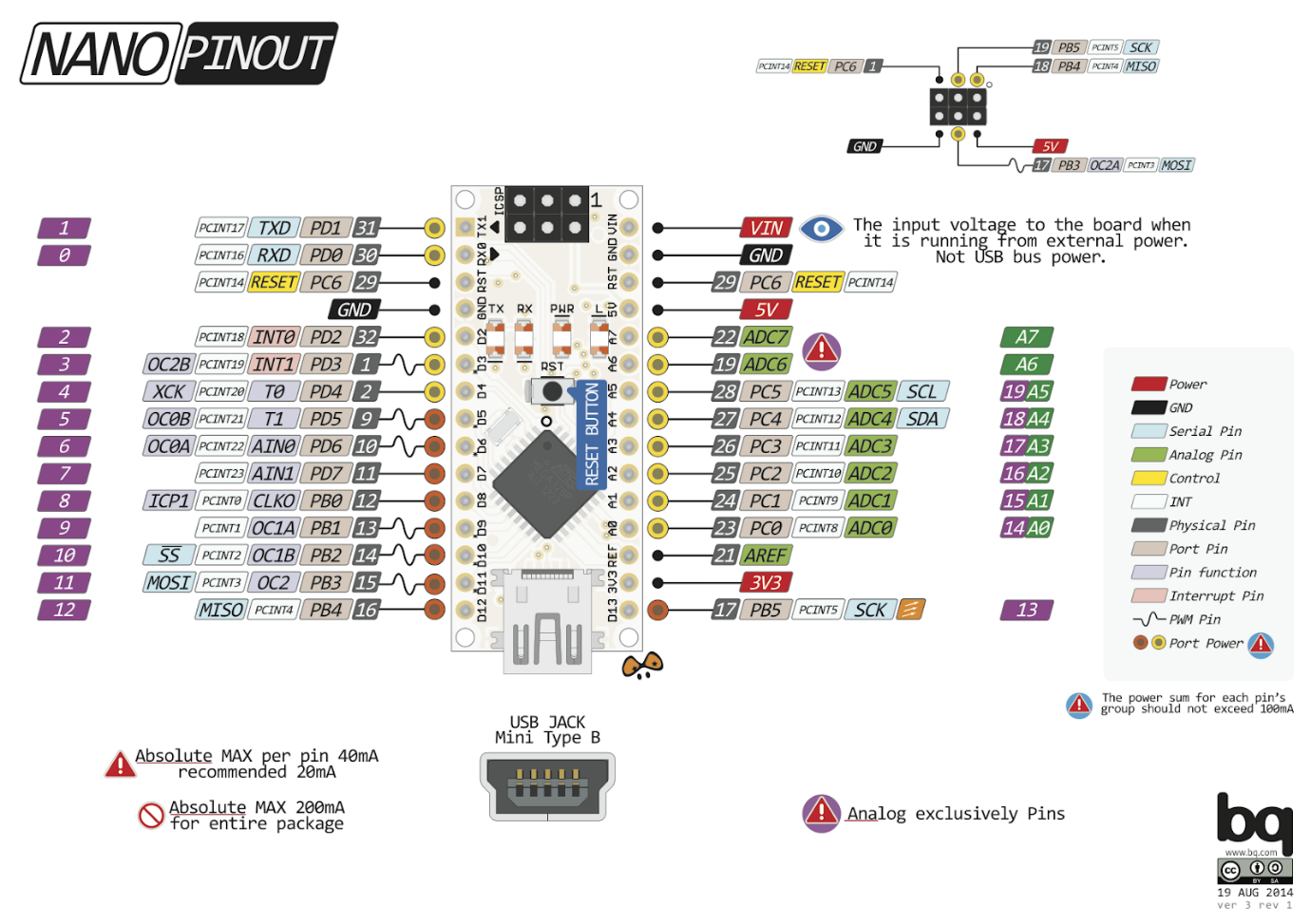
-        Black coloured three pin chips present on the back side of the chip board – Are called MOSFETS – amplifiers used to run motors – takes power in from one pin and controls the output with another – current limiters for MOSFETs included

-        Two orange coloured components on the back of the board – Tantalum capacitors - Filter out noise

-        Free wheeling Diode – Mitigates back emf so that signals are not affected adversely by it

-        LED resistors (LEDs produce light as a function of the activity of resistors) (labelled on the back of the chipset as R11 , R12, R13) which control RGB Parameters to produce different colours on the LED.

-        Check out the Arduino pin map of the chipboard : MPU6050, which is a 32 pin chip; out of these 32 pins - neglecting power supply, crystal control, reset – pin number 0 through 13 are digital I/O pins.



-        UART pin – Universal Asynchronous Receiver Transmitter – This pin is used to programming the chip via USB.

-        Red programming cable to be connected to pin numbers 0 and 1 – other two being intended for power supply

-        Pin 2 – 3 – Digital I/O pins

-        A computer without a bootloader is as good as nada – Change the motherboard if you want it to see another sunrise

-        ICSP – In System Circuit Programmable

-        A0 to A7 – Analog Inputs

-        A0 to A5 – Additional digital I/O features

-        ADC – Analog to digital converter – Capacity = 10 bytes each

-        20 digital I/O pins in total

-        PWM pins – motor connected these for voltage control – 3, 5, 6, 9, 10, 11

-        LED – Front left – Pin 12; Front Right – Pin 11; Back Right – Pin 8; Back Left – 7, R LED - A1, G LED – A0, B LED – A2

-        Motor pins – 5, 6, 9, 10

-        Digital I/O pins can do almost everything by virtue of being versatile

-        Android app to be provided – Detects the drone’s WiFi hotspot for exclusive control of the drone

-        Tool Settings

o   Board – ATMEGA328

o   Bootloader – Yes

o   BOD – Disabled (apparently a flawed redundancy in Arduinos

o   Clock Speed – 8 MHz external

o   Programmer - AVRSP mkll

o   Variant – 328P / 328PA

o   Compiler LTO – (disabled)

-        Let’s get to the code.

-        USB Module Instructions (red chip)

o   Connect the four provided pin cables to pins labelled Ground, RX, TX and 5 volt

o   Connect the USB Module to your PC and check for its driver in Device Manager (Silicon Labs CP210x) (At this point the RGB LEDs will glow one after the other in a clockwise or anticlockwise fashion, due to the code).

o   Last setting – in the Arduino software, for the Device ID; go to ports and check for the comm no.

o   Connect the other ends of the pin cable connected to Ground, RX, TX and 5 volt to pins on the top end of the chipboard (held such that the battery supply port, or whatever that is, is at the bottom of the chip layout)– bottom most to Ground, top most to 5 volt, TX (Transmitter) to the 2nd one from the bottom, RX (Receiver)

-        Before uploading the code from Arduino IDE onto the chip via USB, press and hold the reset button on the flight controller chipset (i.e. the chipset we have been working with) and press upload in the IDE interface; and as soon as the prompt/message uploading shows up on the IDE window, release the button pronto (if there is a problem with infinite uploading, disconnect the USB Module from the PC to force stop the process and then reconnect it and try again).

- Code Written –  
o void setup()   
o {  
o  // put your setup code here, to run once:  
o  pinMode(12,OUTPUT); //pinMode is a keyword (two word function; functions usually use this letter case convention //Using pin 12 and thus the LED connected to it  
o      
o }  
o   
o void loop()   
o {  
o  // put your main code here, to run repeatedly:  
o  digitalWrite(12,HIGH);   /\*digitalWrite - send data; digitalRead - receive data; HIGH - 1 (ON), LOW - 0 (OFF)\*/  
o  delay(100);  //like the sleep function in C++ (parameter in milliseconds)  
o  digitalWrite(12,LOW);  
o  delay(100);  
o } //makes the front left LED connected to pin 12 glow and extinguish repeatedly

-        Additional Task – Write a code to make the LEDs glow sequentially in a clockwise or anticlockwise pattern (using the LED pin numbers already provided) in the IDE.

-

* White wire at right side clockwise
* White wire at the left side anticlock wise

|  |  |  |  |
| --- | --- | --- | --- |
| Part | Right | Left | ref |
| Arm | Cw | Acw |  |
| Motor | Cw | Acw |  |
| propeller | R | L |  |

* The quadcopter ought to be programmed with atmel studios but that would be highly complicated program. The program is made compatible with Arduino c++.

|  |  |  |  |
| --- | --- | --- | --- |
| STATE | W(5) | B(6) |  |
| OFF | 0(LOW) | 0(LOW) |
| CW | 0 | 1 |
| ACW | 1 | 0 |
| OFF | 1 | 1 |

* Analogue method

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Method 1 (2 wire) | |  | Method 2(1 wire) | |
| **5** | **6** |  | **5** | **Ground** |
| 0-255 | 0 | Cw | 0-255 | Gnd |
| 0 | 0-255 | Acw |  |  |
|  |  |  |  |  |

* How to interpret the ESP1266 and how to calibrate
* The standard baud rate for the ESP8266 is 115200.
* AT command means attention command.
* **Serial** is the ESP. mySerial is to connect with the ESP.
* after uploading the step 3 program remove the +5V supply and connect with the battery.