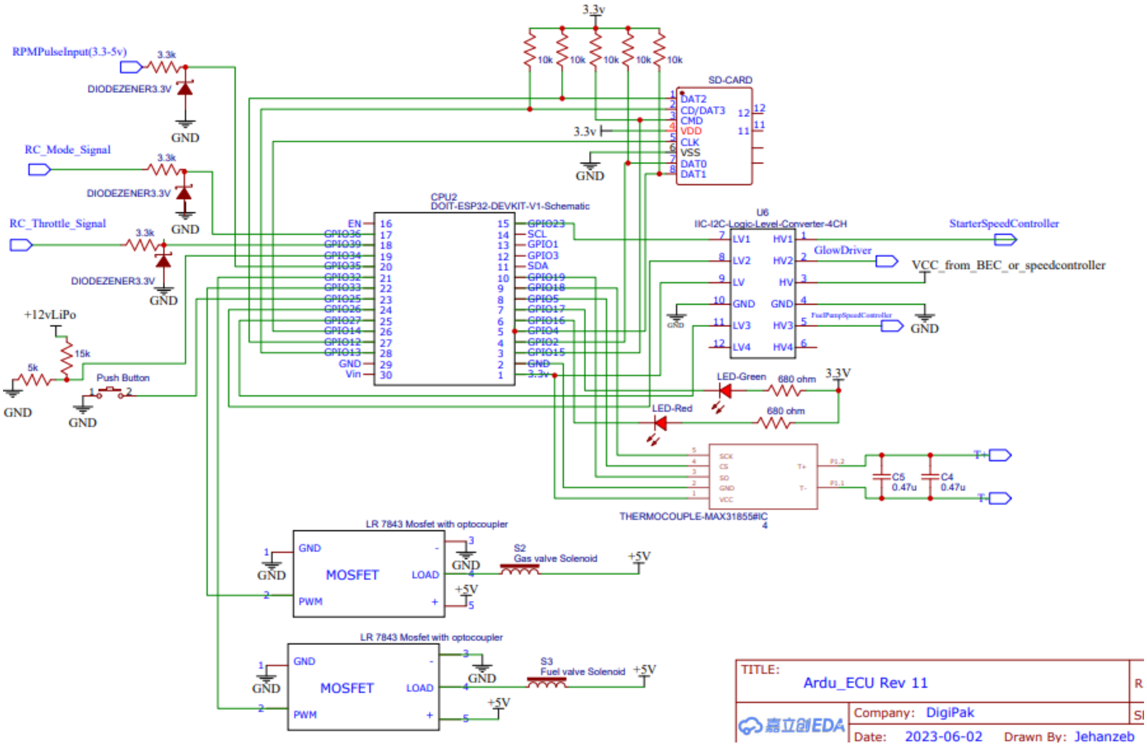
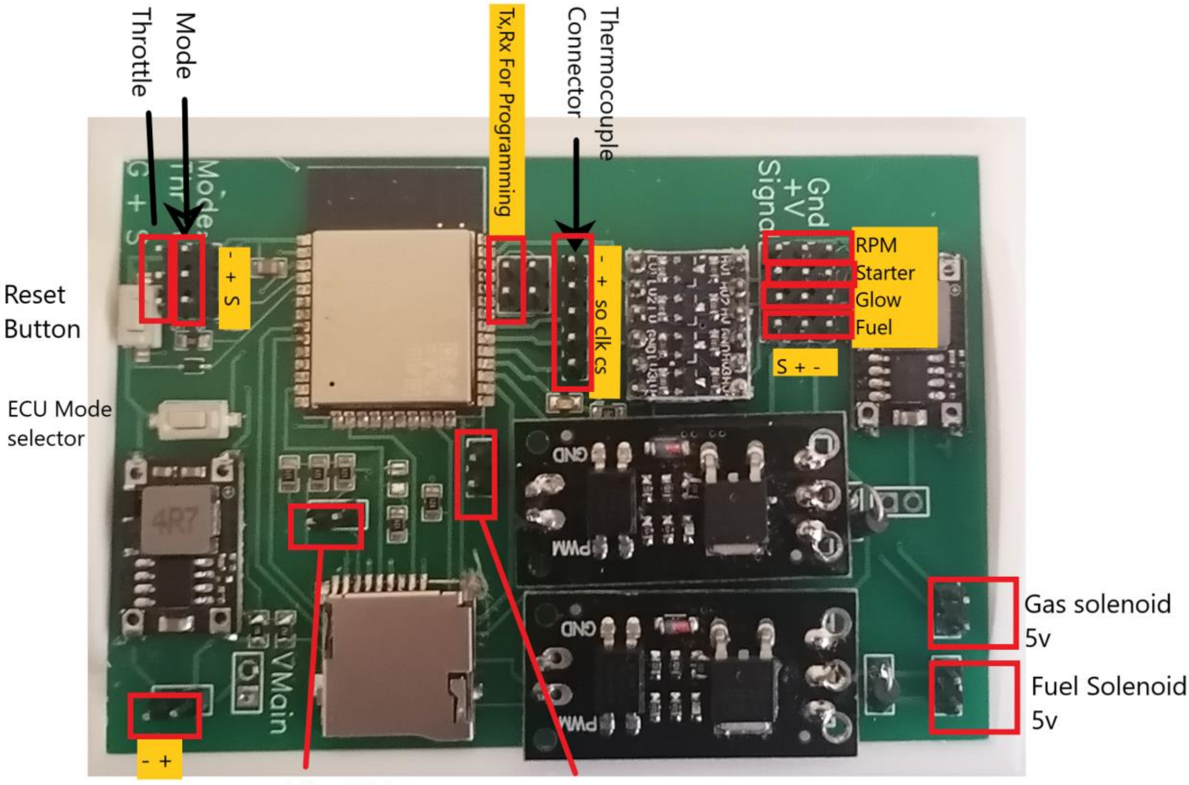
Ardu_ECU Hardware



Ardu ECU can be run on simple hardware based on ESP32 Devkits with dual core ESP32 chips. A sample schematic is provided in github which can be followed to develop custom hardware.

In addition DigiPak has also developed a board with integrated components for people who want a readymade clean solution. The board can be ordered at http://digipak.org



Power Input 6-15 V

Micro SD Jumper Must be shorted for SD card to work

Programming Jumper. Must be shorted while pressing the reset button to enter programming mode

Programming the Rev 11 Firmware on to ESP32

ECU Rev 11 is different from previous versions as it uses specific pins to connect to SD card at high speed in 4 wire mode. The pins used for SD Card are as follows;

GPIO14	CLK
GPIO15	CMD 10k pullup in SD mode
GPIO2D0	10k pullup in SD mode, pull low to go into download mode
GPIO4D1	10k pullup in 4-line SD mode
GPIO12	D2 10k pullup in 4-line SD mode (see Note about GPIO12 below!)
GPIO13	D3 pin must have a 10k pullup

Behavior of two pins require careful attention in this configuration

GPIO2

GPIO2 pin is used as a bootstrapping pin, and should be low to enter UART download mode. One way to do this is to connect Ground and GPIO2 using a jumper, and then reset the board. This will cause the board to enter programming mode and code can be uploaded to ESP32 board. After programming, this jumper has to be removed for ESP32 and a reset performed to be able to run the code. If not done, ESP32 will remain in programming mode

GPIO12

GPIO12 is used as a bootstrapping pin to select output voltage of an internal regulator which powers the flash chip. This may be incompatible with some ESP32 modules and will not allow SD card to work. To get around this problem you have to enter in programming mode and use the following command

```
components/esptool_py/esptool/espefuse.py set_flash_voltage 3.3V
```

After this the ESP32 Board can be programmed normally.

Detail of this issue can be found at following link

https://github.com/espressif/esp-idf/tree/master/examples/storage/sd_card/sdmmc#4-line-and-1-line-sd-modes

Connecting to User interface

Ardu_ECU Rev11 is based on a Web based user interface similar to previous versions.

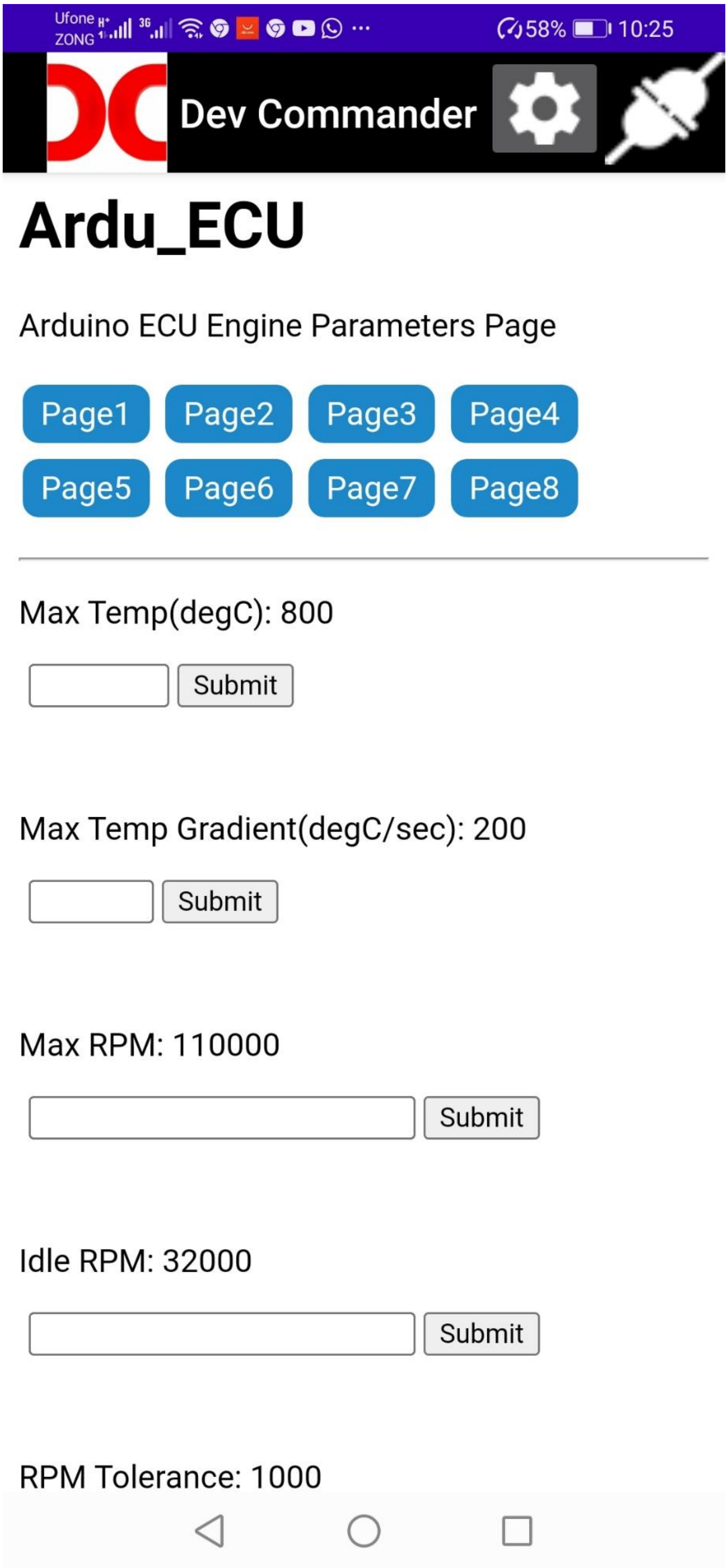
ECU provides a WiFi WebServer with default name as "Ardu_ECU" and default password "admin123".

Both these values can be changed through user interface for privacy and security

After connecting to Ardu_ECU wifi connection, the initial page can be opened by typing http://192.168.4.1 in a web browser.

Alternately an Android app "DevCommander" is also developed in which WiFi SSID, Password and page address is entered and the app can be used to open the ECU webpage. Using the app makes it easy to connect to ECU. It must be noted that if the app is used, the Ardu_ECU WiFi configuration should not exist in Android remembered networks

Page1



Ardu_ECU

Arduino ECU Engine Parameters Page

Max Temp(degC): 800

Max Temp Gradient(degC/sec): 200

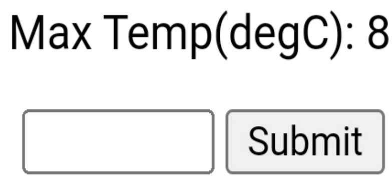
Max RPM: 110000

Idle RPM: 32000

RPM Tolerance: 1000

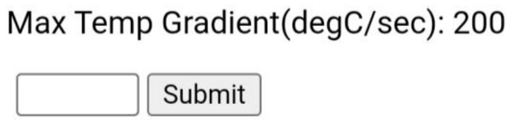
Page 1 allows user to change various settings to control the ECU.

The settings that can be changed on Page1 are as follows;



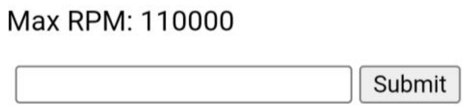
Max Temp(degC): 800

Max Temp Sets the maximum temperature limit. When the temperature exceeds this value the ECU reduces the fuel flow. This means that temperature can exceed maxTemp however when this happens the ECU will react to correct the situation. As most of the turbines are made of Inconel alloy. A max temperature of 800 degC is recommended. Short exposures to higher temperatures at low RPM might not affect the engine but at high RPM and 10 of seconds of exposure are likely to damage the engine above this temperature.



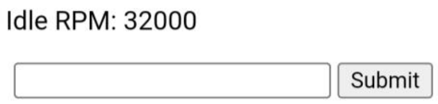
Max Temp Gradient(degC/sec): 200

Max Temp Gradient limits how quickly the temperature is allowed to change. When the rate of change of temperature exceeds this value the ECU stops increasing fuel flow.



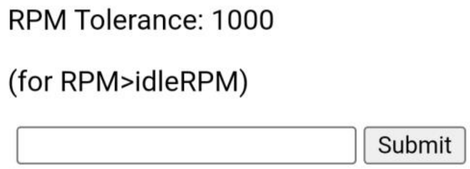
Max RPM: 110000

Max RPM sets the maximum RPM limit. When the RPM exceeds this value the ECU reduces the fuel flow. This means that RPM can exceed maxRPM however when this happens the ECU will react to correct the situation. Max RPM for most engines is in 115000 to 125000 RPM range and 110000 RPM Limit provides some safety margin.



Idle RPM: 32000

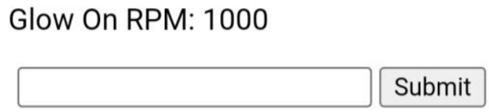
Idle RPM sets the target RPM that engine holds after startup. When the engine is in auto start mode and the mode switch is flipped to perform autostart, the ECU will go through the startup steps and once engine is above self-sustaining RPM, it will hold the Idle RPM. It is a good practice to set at least Idle RPM3000-4000 RPM above self-sustaining RPM



RPM Tolerance: 1000

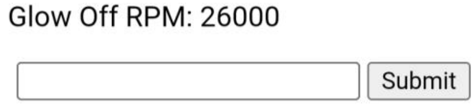
(for RPM>idleRPM)

RPM Tolerance sets the range of RPM that engine holds. When the engine is in auto start mode and the mode switch is flipped to perform autostart, the ECU will go through the startup steps and once engine is above self-sustaining RPM, it will hold the Idle RPM. It is a good practice to set a wide RPM tolerance of about 5000 RPM to avoid ECU chasing a very tight RPM range and cause repeated acceleration and deceleration.



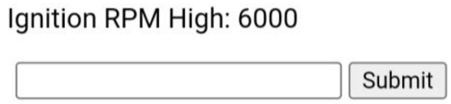
Glow On RPM: 1000

This is the RPM above which, the glow plug will be switched on. This should be lower than Gas On RPM so that the gas is immediately ignited once the gas flow is switched on.

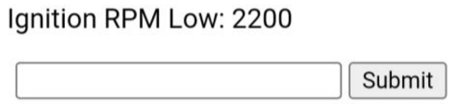


Glow Off RPM: 26000

Glow plug will be switched off once the RPM exceeds Glow off RPM. Ideally this should be between starterOffRPM and idleRPM.

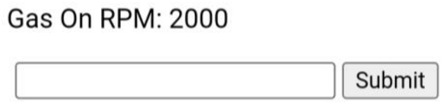


Ignition RPM High: 6000



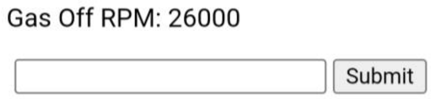
Ignition RPM Low: 2200

Ignition RPM High and ignition RPM Low. These RPM set the limits to enable initial ignition of gas inside the jet engine. During the startup the starter spins the engine until it reaches Ignition RPM High. At this point starter is switched off and RPM is allowed to die down. As it reaches Ignition RPM Low, the starter spins again to raise RPM to Ignition RPM High. During this process the gas is also switched on according to gasOnRPM as well as glow Plug is turned on if RPM is above glow On RPM. The change in RPM as engine slows down or speeds up between Ignition RPM High and Ignition RPM Low changes the gas air mixture. This coupled with heat from glow plug creates the right conditions for ignition. The engine will keep oscillating between IgnitionRPMHigh and IgnitionRPMLow until 1)ignition is achieved or 2)Ignition timeout limit is exceeded when the ECU will stop the starting process with a No Ignition error.



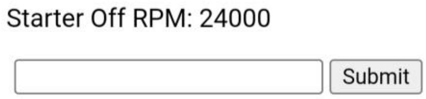
Gas On RPM: 2000

Gas solenoid will be activated and gas flow will be switched on once the engine RPM exceeds this value. It is best to set this value 300-400 RPM below IgnitionRPMLow.



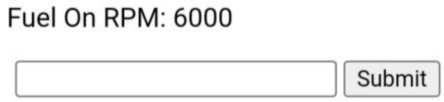
Gas Off RPM: 26000

Gas solenoid will be de-activated and gas flow will be switched off once the engine RPM exceeds this value. It is best to set this value 3000 RPM above starterOffRPM when the engine is above self sustain RPM.



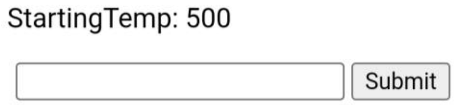
Starter Off RPM: 24000

This is the RPM at which engine is fully self sustaining, starter will be switched Off and engine will be running on fuel only. If starterOff RPM is set too low, the fuel will increase but RPM will not increase as engine is not yet at self sustain RPM. This value depends on engine construction and efficiency of compressor and turbine. For KJ66 type engines, self sustain RPM is around 20000-22000 and for such case starterOffRPM can be set at 23000-24000 RPM. If the engine begins to slow down after starter is switched off then starterOffRPM shall be increased. Starter motor shall also be capable of delivering this RPM as if starter motor cannot reach starter off RPM then the engine will fail to start.



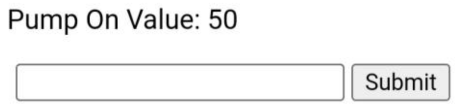
Fuel On RPM: 6000

This is the RPM, below which the fuel pump will not start pumping. It should be set anywhere above ignitionRPMLow to ensure that fuel is only pumped once ignition is established

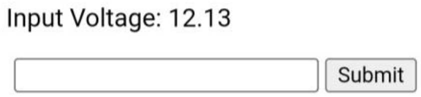


StartingTemp: 500

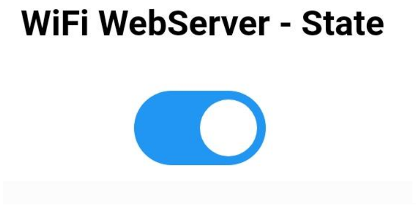
This value defines the starting temperature (degC) of the engine. Starting temperature greatly affects the startup of jet engine. A key role of starter motor after giving the initial spin to engine is to keep exhaust temperature low. Starter does this by comparing exhaust temperature and starting temperature. If exhaust temp is lower than starting temp, starter maintains current RPM and if exhaust temp is higher than starting temp then starter speeds up to introduce more air into the engine and cool the exhaust. If Starting temp is kept too low , it is possible to extinguish the flame in engine and cause a flameout abort during start. On the other hand if this temperature is kept too high, it might be possible for engine to exceed maximum temperature as starter will not have enough time to react to increased fuel flow. Default value of 500 degC. Has been found suitable for KJ66 engine during testing.

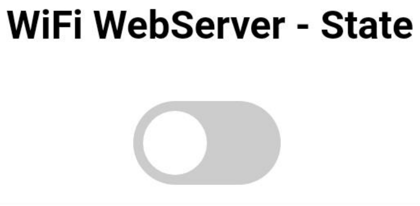

Pump On Value: 50

The speed of pump has a control range of 0-1000 in ECU Rev11. Most pumps will not pump anything at low values and only start pumping once a certain value is exceeded. This “no flow” range only causes a delay in engine startup and makes starter run for longer time. To avoid this, the pump on value can be used. This value has to be determined manually. It can be observed by putting ECU in pump prime mode and determining when exactly the fuel starts flowing through the pump. This value can then be entered and will be used for all subsequent starts.


Input Voltage: 12.13

This is the voltage that is read through a voltage divider to determine the input to ECU. As this is based on nominal ratio of voltage divider network it can be corrected by entering the exact value. The exact voltage can be measured with an accurate voltmeter and entered here to correct the Voltage divider ratio. Note: in this cell the actual voltage needs to be entered, the correct ratio will be calculated by ECU internally.

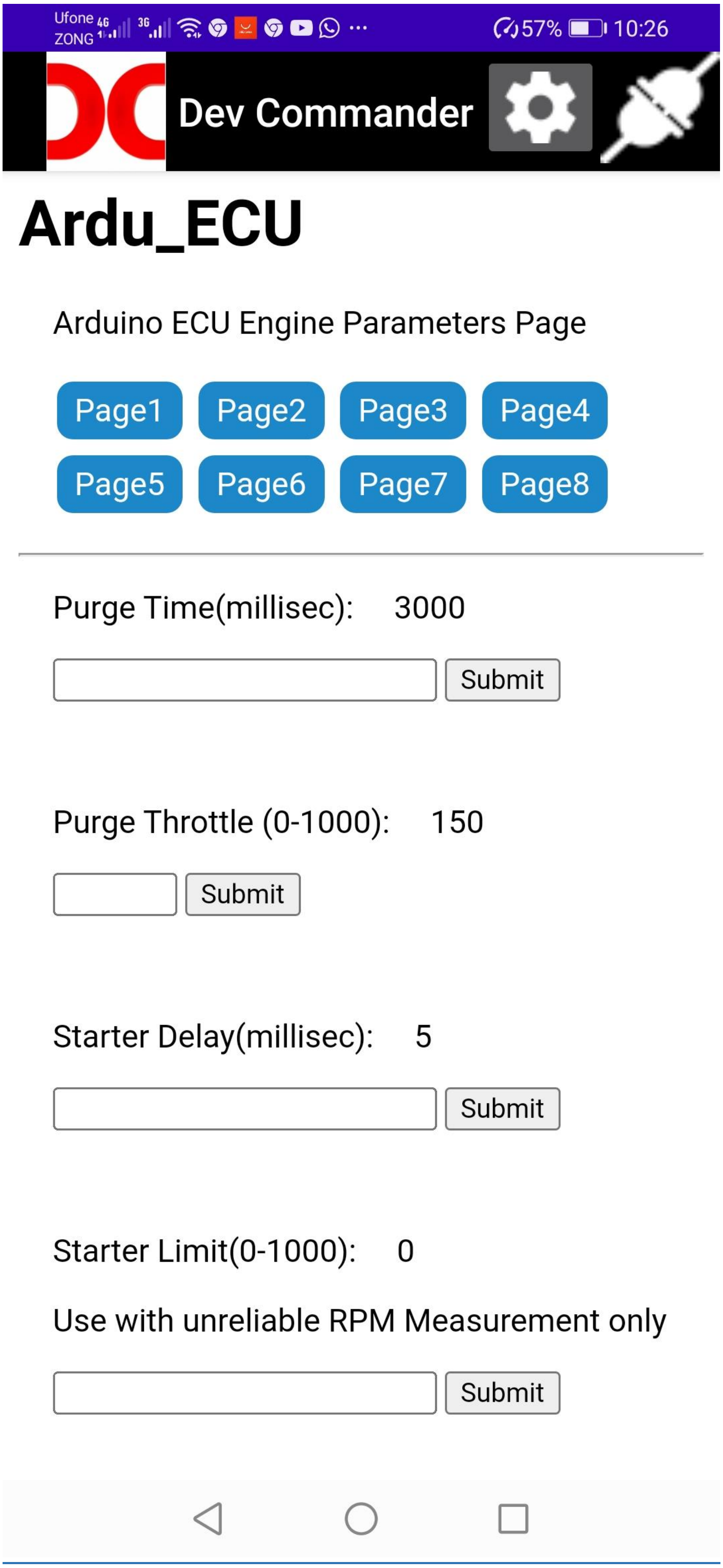

WiFi WebServer - State


WiFi WebServer - State

This switch is used to turn off the Wi-Fi webserver. All functions of ECU will work normally but Wi-Fi interface will be switched off and it will not be possible to change settings or watch sensor values. This is designed so that in a flying RC jet model, one would not want a Wi-Fi transmitter operating next to RC receiver. Other than Wi-Fi, all functions of ECU will perform normally.

Wi-Fi can only be enabled again by cycling the power to ECU.

Page2



Ardu_ECU

Arduino ECU Engine Parameters Page

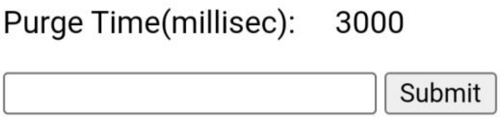
Purge Time(millisec): 3000

Purge Throttle (0-1000): 150

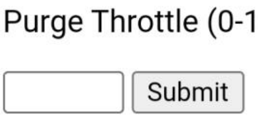
Starter Delay(millisec): 5

Starter Limit(0-1000): 0

Use with unreliable RPM Measurement only

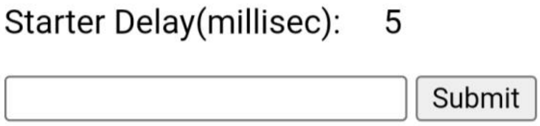


Purge Time(millisec): 3000



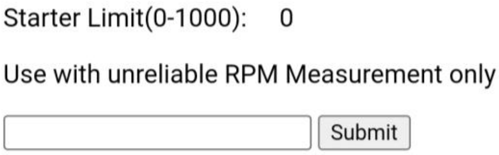
Purge Throttle (0-1000): 150

At the beginning of startup cycle ECU performs a purge operation. The ECU spins the engine for time=" Purge Time" at throttle setting of "Purge Throttle". The purpose is to clear engine of any residual gases from previous operation.



Starter Delay(millisec): 5

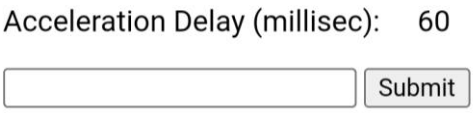
Starter Delay value determines how quickly starter speeds up or slows down. The higher the delay, the slower will be starter response and smaller delay value will cause starter to respond quicker. An extremely fast starter spool up might cause a flameout while an extremely slow spool up might cause high exhaust temperature and damage the turbine. Default of 5 ms has been optimum for KJ66 engine.



Starter Limit(0-1000): 0

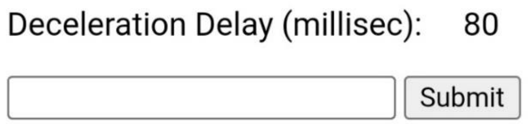
Use with unreliable RPM Measurement only

Starter limit should only be set if the RPM measurement is not reliable during startup but is good when starter motor is off. This situation arises when starter motor generated electromagnetic noise affects the hall type RPM sensors and RPM measurement is affected. When Starter Limit is set at anything other than zero, the startup sequence ends when the starter reaches this limit. ECU does not look at RPM, it only increases the starter throttle and Fuel flow according to set parameters. Once starter reaches the limit value, the ECU assumes that the engine is self-sustaining and will switch off the starter. For all cases where RPM measurement is good this value should be zero.



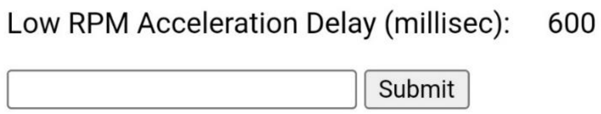
Acceleration Delay (millisec): 60

Based on current parameters and the user input, the ECU firmware calculates if the fuel pump output needs to be increased. This calculation is performed thousands of times a second but fuel must be increased gradually to avoid pumping excess fuel in the engine. Acceleration delay determines the interval between each increase of fuel pump. The lower this value, the faster the fuel pump will increase. It must be mentioned that most of the calculations are based on Temperature measurement of exhaust gases and this measurement cannot be performed faster than 100milliseconds based on Max31855k Thermocouple chip. This means any value of acceleration delay faster than 100ms is just based on old results and is not based on new information. Also, acceleration delay is applied when engine RPM> Idle RPM.



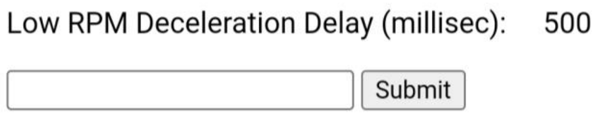
Deceleration Delay (millisec): 80

Deceleration delay determines the interval between each reduction of fuel pump output. The lower this value, the faster the fuel pump output will decrease. It must be mentioned that most of the calculations are based on Temperature measurement of exhaust gases and this measurement cannot be performed faster than 100milliseconds based on Max31855k Thermocouple chip. This means any value of deceleration delay faster than 100ms is just based on old results and is not based on new information. Also, deceleration delay is applied when engine RPM> Idle RPM.



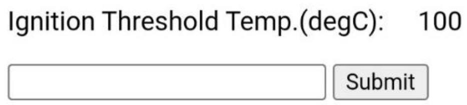
Low RPM Acceleration Delay (millisec): 600

Same as acceleration delay however this delay is applied only when engine RPM<idleRPM.



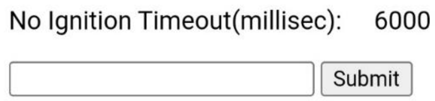
Low RPM Deceleration Delay (millisec): 500

Same as deceleration delay however this delay is applied only when engine RPM<idleRPM.



Ignition Threshold Temp.(degC): 100

This is the temperature above which the ECU will consider that ignition has occurred and move on to next steps in starting the engine. Also, if an abort condition has happened, the exhaust temperature needs to be dropped below ignition threshold to begin the startup sequence again.



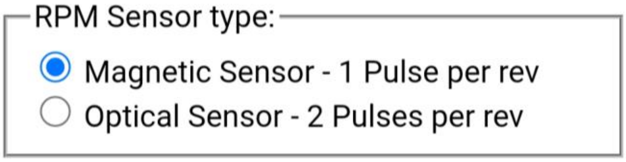
No Ignition Timeout(millisec): 6000

During startup , the ECU will switch on Glow Plug and turn on gas valve and oscillate between ignitionRPMHigh and ignitionRPMLow. This process will continue until one of the two things happen

1)ignition happens and temperature rises above IgnitionThreshold at which point the ECU moves to next stages of starting

OR

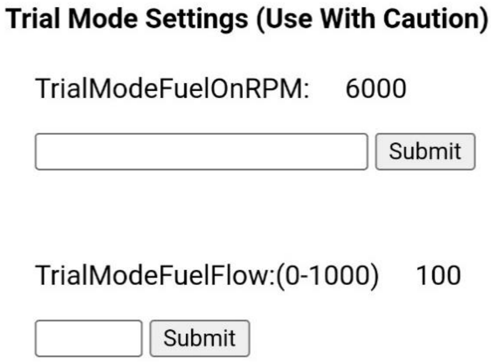
2) No Ignition timeout is exceeded in which case the startup is aborted with a “no ignition” error



RPM Sensor type:

● Magnetic Sensor - 1 Pulse per rev
○ Optical Sensor - 2 Pulses per rev

For RPM measurement the ECU expects a square wave with 1 or two pulses per revolution.There are two types of RPM sensors in common use. Usually Hall type magnetic sensors will output 1 pulse for each revolution of Compressor nut while some magnetic and optical sensors will output two pulses for each revolution. Select the right value carefully as it can half the measured RPM value and lead to dangerous RPM during operation.



Trial Mode Settings (Use With Caution)

TrialModeFuelOnRPM: 6000

TrialModeFuelFlow:(0-1000) 100

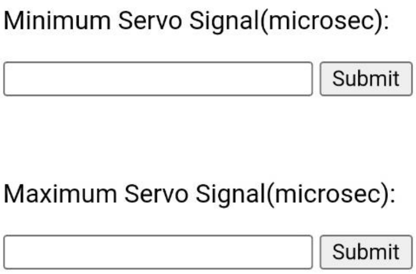
Ardu_ECU provides three main modes of operation for a jet engine. 1) Normal (Auto-Start) 2)Trial Starter Control and 3) Trial Fuel Pump control.

With the proper settings engine should always be run in Normal(Auto-Start) mode. In this mode the engine starts up by flipping the mode switch and reaches idle RPM. From there onwards, the throttle input controls the engine RPM upto maximum of maxRPM.

The Trial Mode Settings apply to following two modes;

2) Trial Starter Control. The throttle stick controls the starter motor only and mode switch controls the gas flow only. The glow plug is switched on as per RPM settings. When the ECU detects ignition and RPM above TrialModeFuelOnRPM, it will start increasing fuel flow and keep increasing it till TrialModeFuelFlow is reached. It is therefore important to set TrialModeFuelFlow to minimum required as Fuel flow will keep increasing till this limit and cannot be controlled by throttle stick.

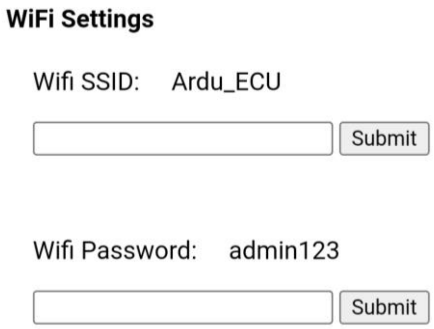
3)TrialFuelPumpControl. The throttle stick controls fuel pump from 0-TrialModeFuelFlow. The mode switch will start the startup sequence and control everything except the fuel pump. Fuel pump control will be via throttle stick but limited to maximum value of TrialModeFuelFlow.



Minimum Servo Signal(microsec): 1000

Maximum Servo Signal(microsec): 2450

This will set the servo signal limits for OUTPUT Only. It has no affect on input range.



WiFi Settings

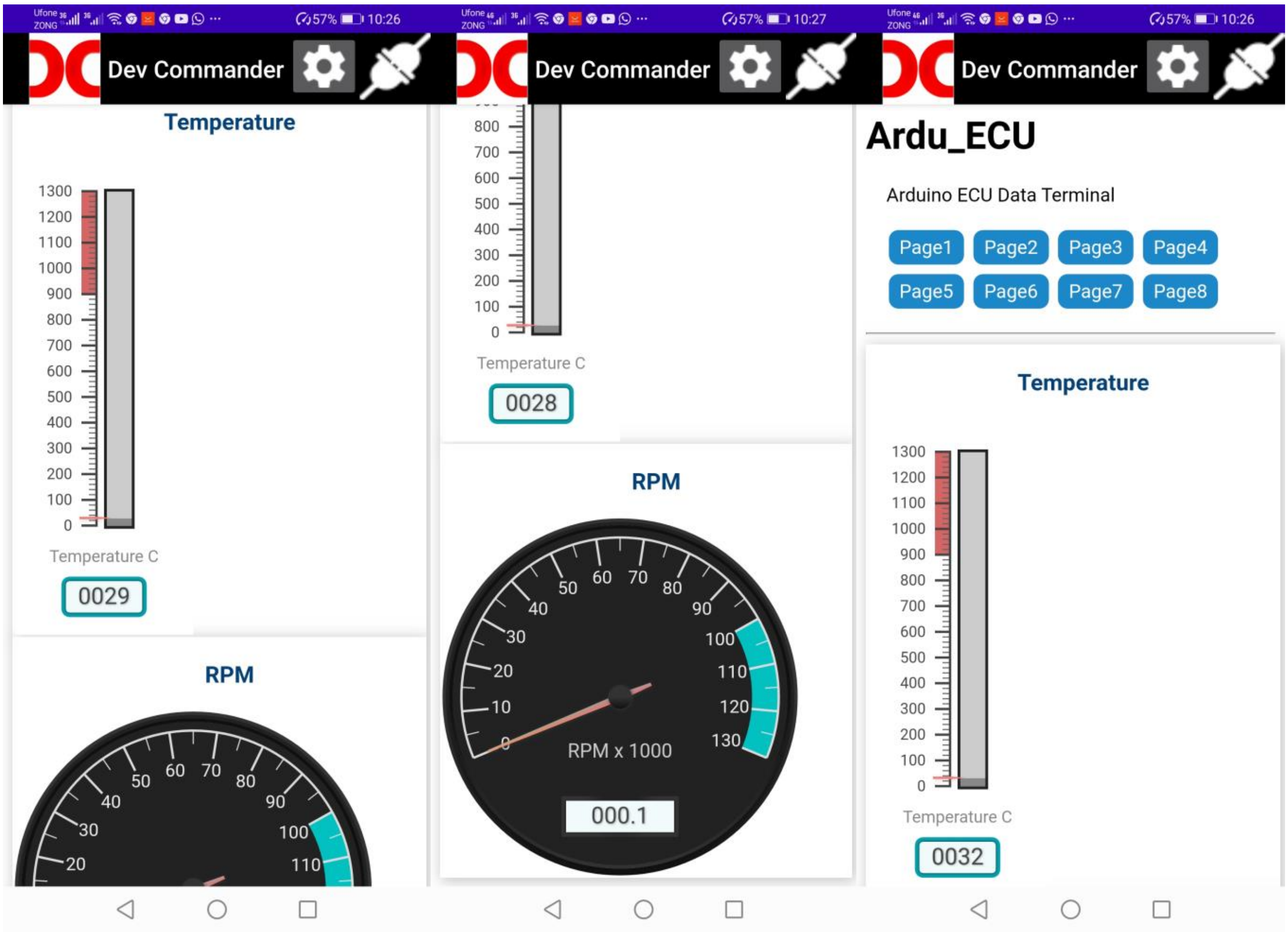
Wifi SSID: Ardu_ECU

Wifi Password: admin123

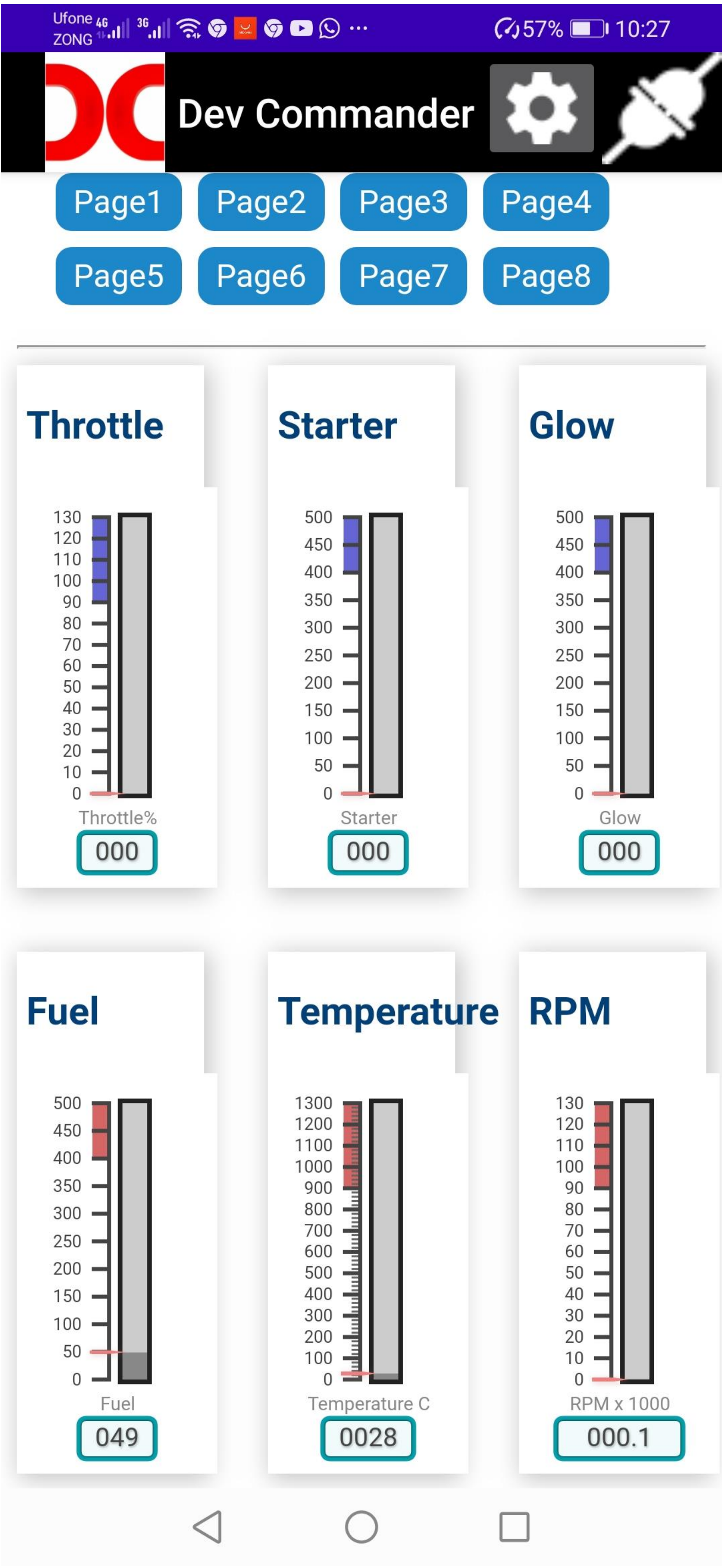
These settings can be used to change WiFi SSID and password to avoid conflict with other Ardu_ECU users. WiFi SSID is also used as a folder name to store engine run data on sd card

Page3

RPM and Temperature gauge

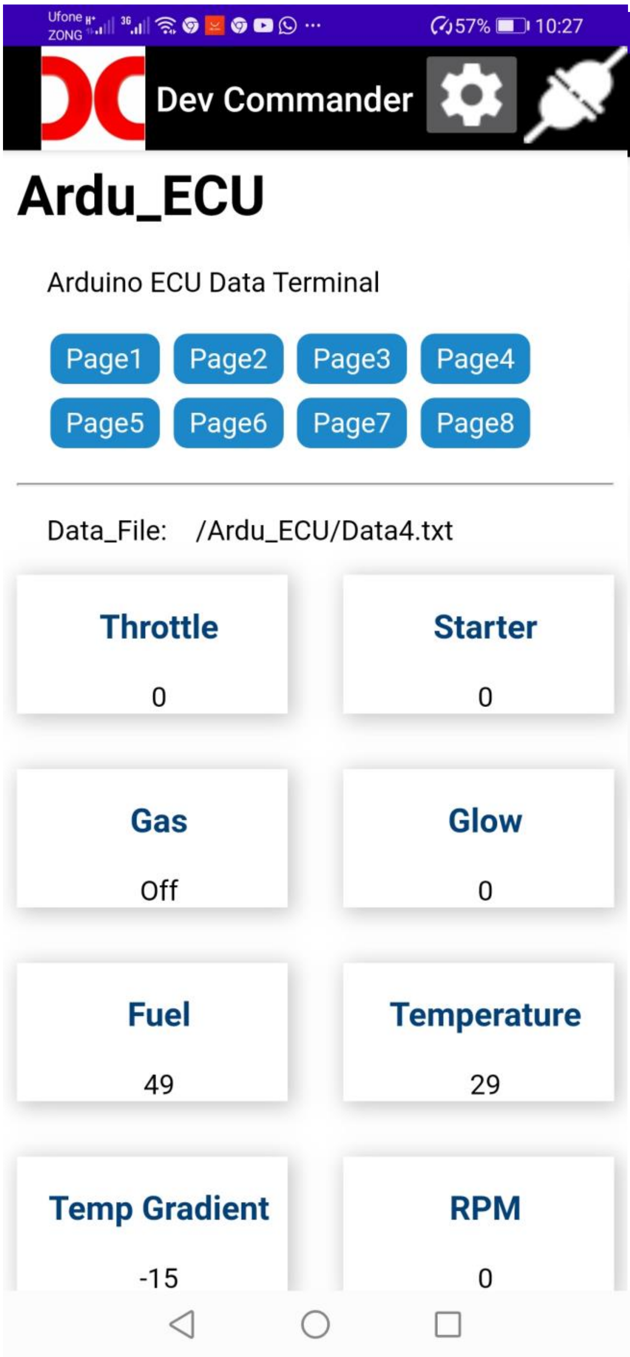


Page 4



More gauges

Page 5

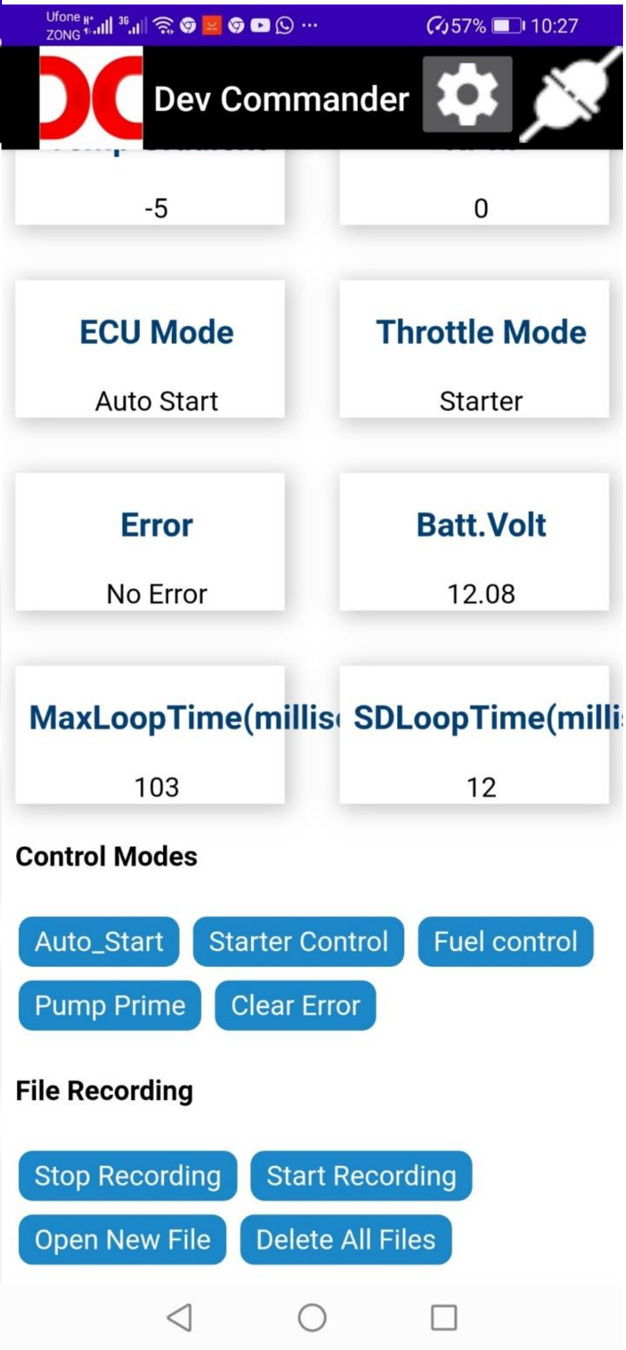


Ardu_ECU

Arduino ECU Data Terminal

Data_File: /Ardu_ECU/Data4.txt

Throttle 0	Starter 0
Gas Off	Glow 0
Fuel 49	Temperature 29
Temp Gradient -15	RPM 0



-5	0
ECU Mode Auto Start	Throttle Mode Starter
Error No Error	Batt.Volt 12.08
MaxLoopTime(millis 103	SDLoopTime(milli 12

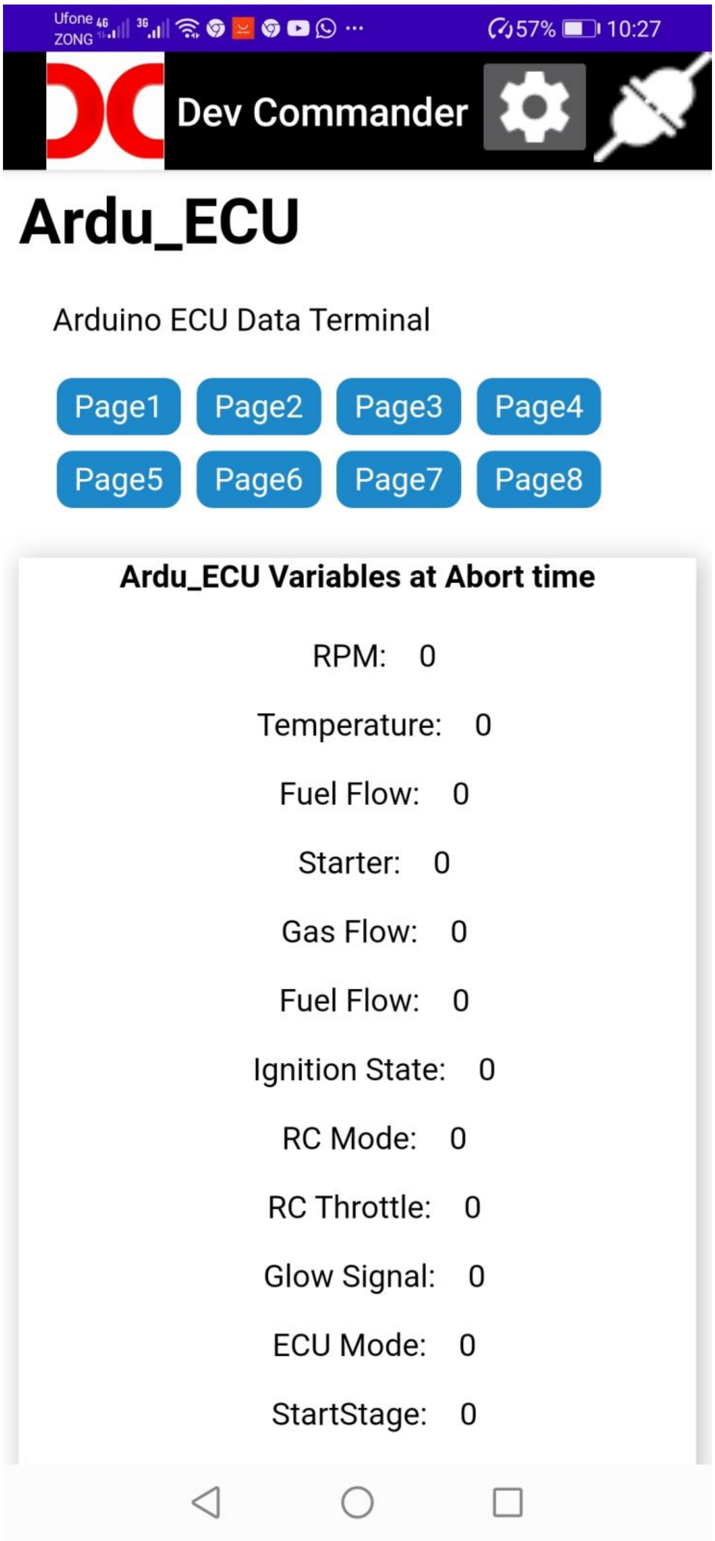
Control Modes

File Recording

Page 5 Provides Value display of various inputs and outputs. It also shows the name of data file where all the data is being recorded.

At the lower part of page buttons are provided to control Engine modes and perform file operations

Page 6



Ardu_ECU

Arduino ECU Data Terminal

Ardu_ECU Variables at Abort time

RPM: 0

Temperature: 0

Fuel Flow: 0

Starter: 0

Gas Flow: 0

Fuel Flow: 0

Ignition State: 0

RC Mode: 0

RC Throttle: 0

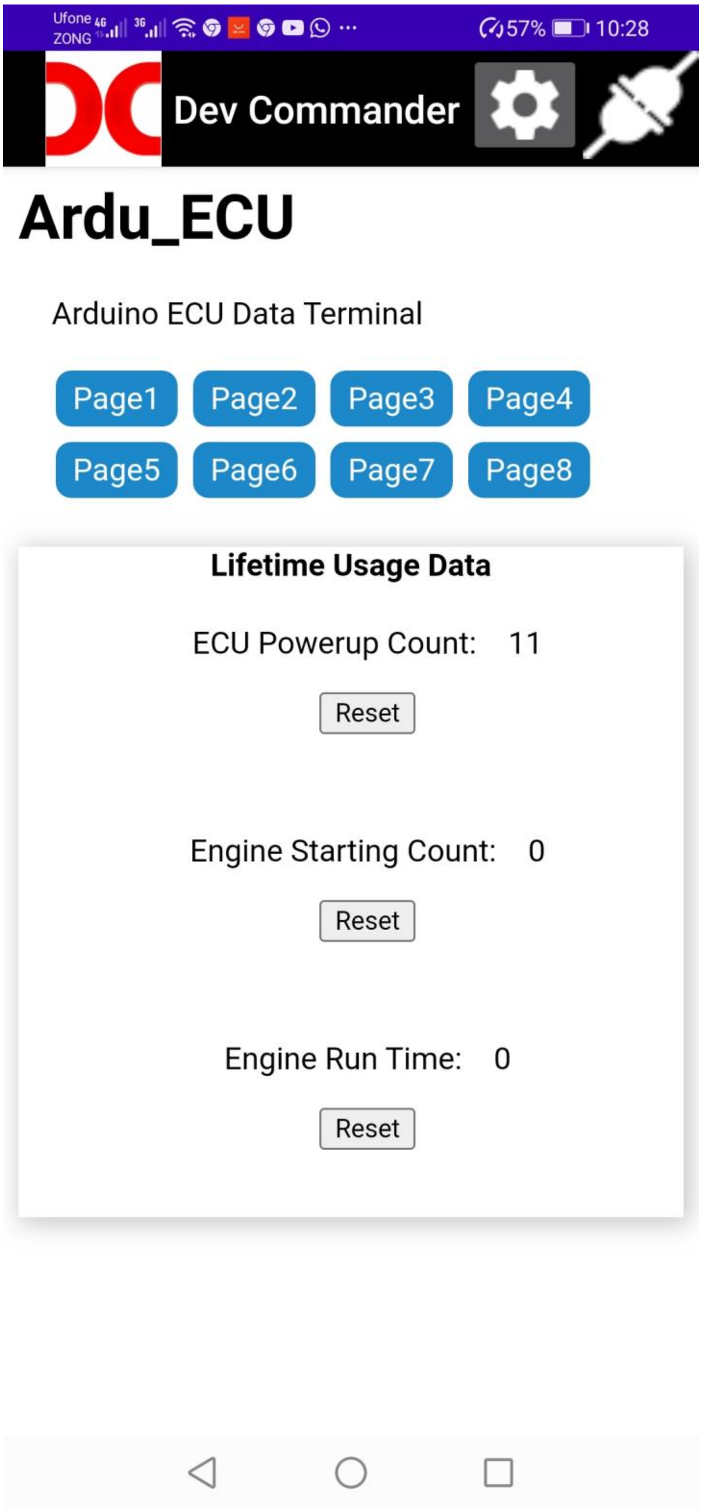
Glow Signal: 0

ECU Mode: 0

StartStage: 0

Page 6 displays a snapshot of various parameters at the last abort event.

Page 7



Ardu_ECU

Arduino ECU Data Terminal

Lifetime Usage Data

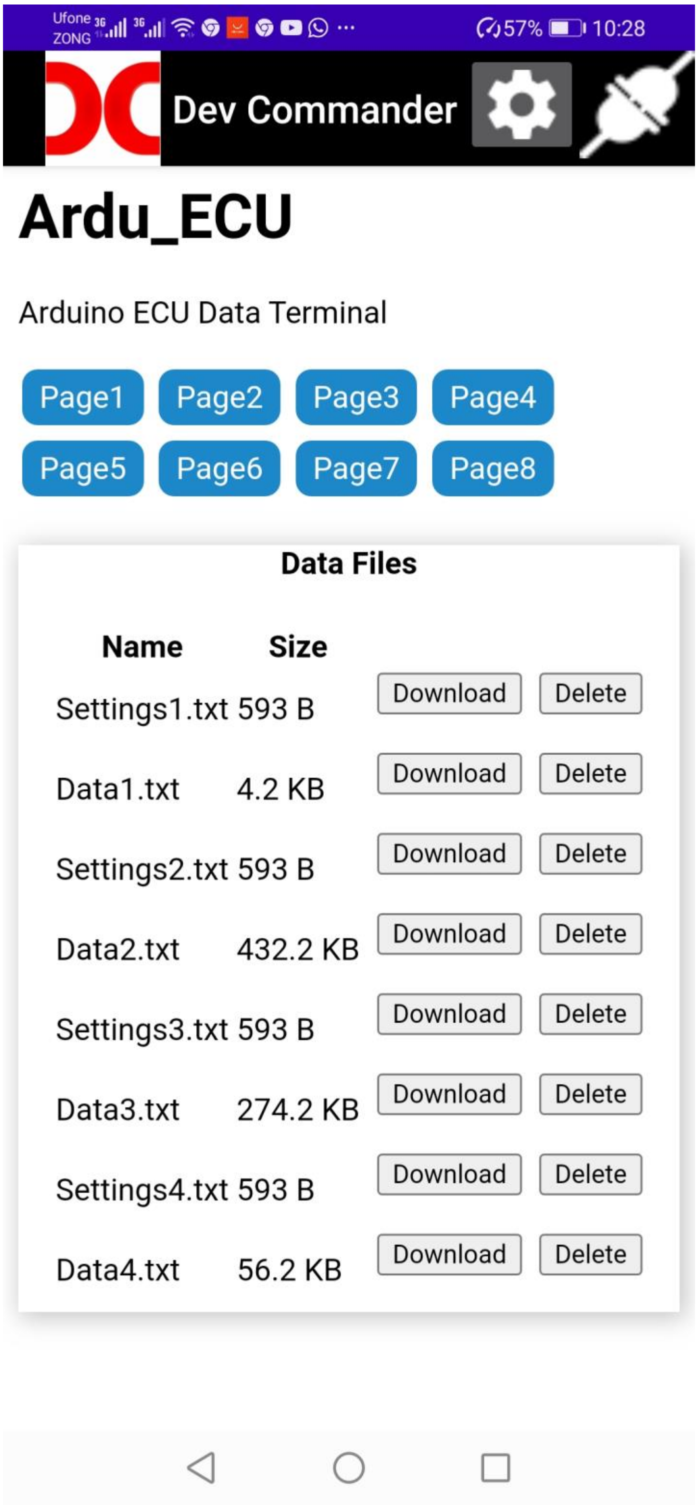
ECU Powerup Count: 11

Engine Starting Count: 0

Engine Run Time: 0

Page 7 displays engine and ECU start count and engine Run time in minutes

Page 8



Ardu_ECU

Arduino ECU Data Terminal

Data Files

Name	Size		
Settings1.txt	593 B	Download	Delete
Data1.txt	4.2 KB	Download	Delete
Settings2.txt	593 B	Download	Delete
Data2.txt	432.2 KB	Download	Delete
Settings3.txt	593 B	Download	Delete
Data3.txt	274.2 KB	Download	Delete
Settings4.txt	593 B	Download	Delete
Data4.txt	56.2 KB	Download	Delete

Page 8 displays all the files recorded by ECU and the option to download or delete these files