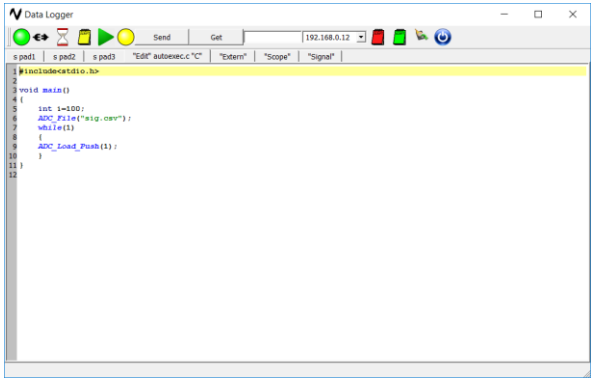
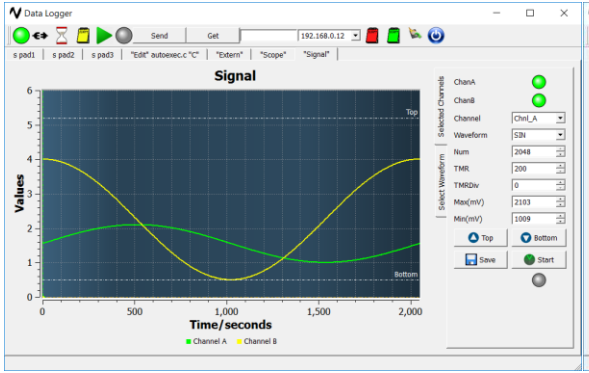
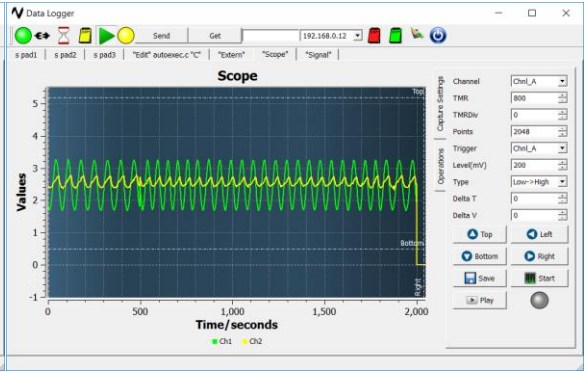
Small Task Prototype System V1

Quick Prototype System

Open source GUI https://github.com/vijayandra/logger_v1







Modify according to need and plan to start and support for for creating open source Android/MAC

Concept

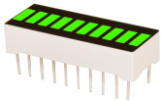
Device **runs user defined “C” code without compiling.** This saves time of learning device specific compiler/debugger/Libraries. STPS system has onboard SD card for saving device specific data.

For example:
“**autoexec.c**” file in SD Card(FAT32) will cause LED to be ON for second and then turned OFF.

```
#include<stdio.h>
void  main()
{
    led_on(1);
    ms_wait(100);
    led_off(1);
    Return 0;
}
```

How much time it will take to run first “C” script

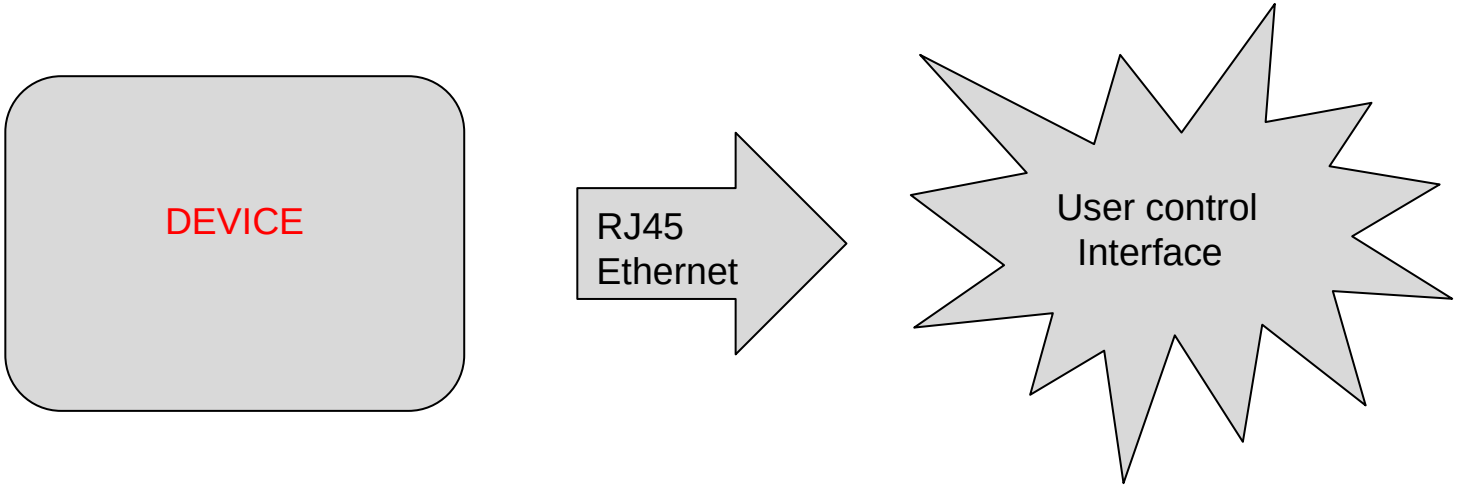
May take less then 5 mins, open box connect to ethernet (Where device can get IP address via DHCP). Start application on any of local network computer Windows PC/Linux PC/or Raspberry PI. Start running supplied “C” scripts.

There are 10 onboard LED  with 1 audible buzzer. Which user can interact with “C” during learning flow and later stage LED/Buzzer can be used for feedback.

```
#include<stdio.h>
void  main()
{  led_on(1);
    buzzer_beep(100);
    ms_wait(100);
    led_off(1);
    Return 0;
}
```

How Application can locate my Device on same network

Device starts broadcasting one packet every 5 seconds interval with its IP address as soon as DHCP allocates IP address. PC application listens to this port for packets and connects to device so user can run script.



Available Resources on Device

1 Ethernet 10/100 Port
2 TTL UART (max 5.65MBPS)
1 RS485 Port with (max 15MBPS) (LTC2855CGN)
1 I2C
10 LED (Intended for active visual feedback)
1 Buzzer (3 KHz audible feedback)
1 SD Card slot (FAT32)
1 Temperature/Humidity Sensor (SI7020)
2 ADC Port Input (Dual 12 bit 3.3V)
2 DAC Port Ouput (Dual 12 bit 3.3V)

Protocol is been used for communication

CoAP protocol encapsulated in RP (header below)is been used for communication, bottom layer of protocol is not part of CoAP standard.Every time when following frame transmitted receiver puts header (8 bytes in black) back to sender, sender marks frame received.

0x55H	0x55H	Len H	Len L	CRC H	CRC L	MID H	MID L	**CoAP**

Autoexec.c/Autoexe.ini

System starts executing autoexec.c file loaded from FAT32 SD Card.

For example following script turns LED off and LED on, at the end it issues audible buzzer

```
#include<stdio.h>
void  main()
{       int i,j;
        for(j=1; j<=9; j++)
        {
            led_on(j);
                    ms_sleep(20);
                    led_on(j);
       }
       buzzer_on(300);
      Return 0;
}
```

Possible/Intended Use cases.

System can accept memory card upto 32G SD card.

A. Remote/autonomous data logging for UART/RS485/Analog/Digital

B. Home Temperature/Humidity logging directly into SDIO card.

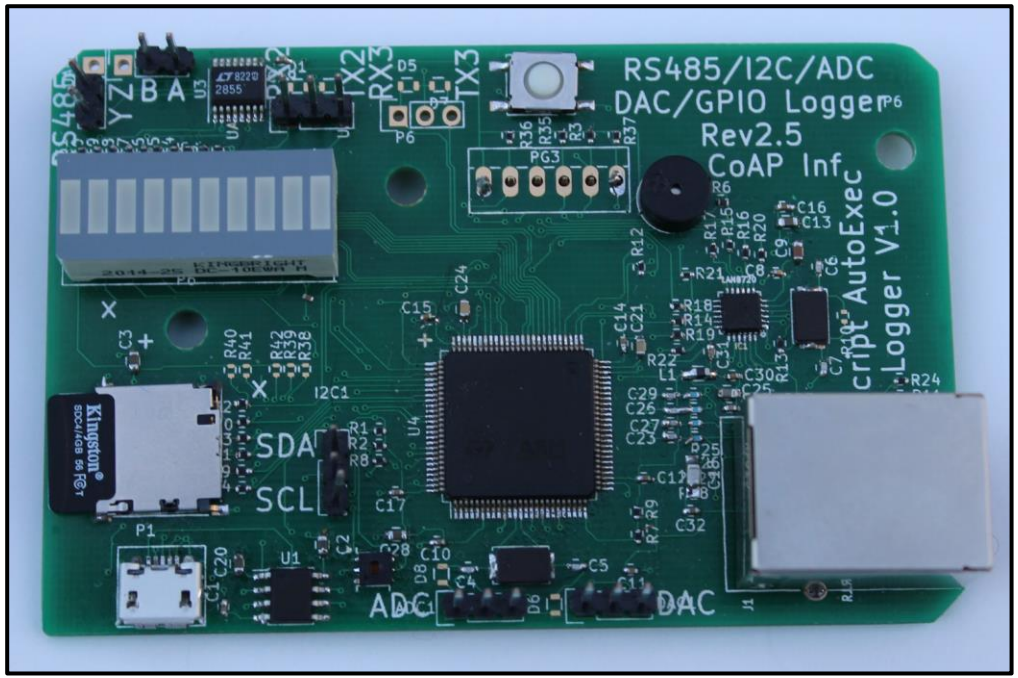
C. Quick turnaround rapid prototype development system.

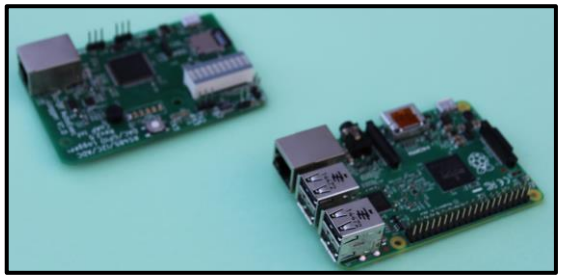
D. Teaching assistance with “C”programming hardware

E. Close loop automated testing platform.

Reuse Raspberry PI2 Enclosure

A. Credit card size board just like raspberry PI, it may be accommodated in raspberry PI enclosure.

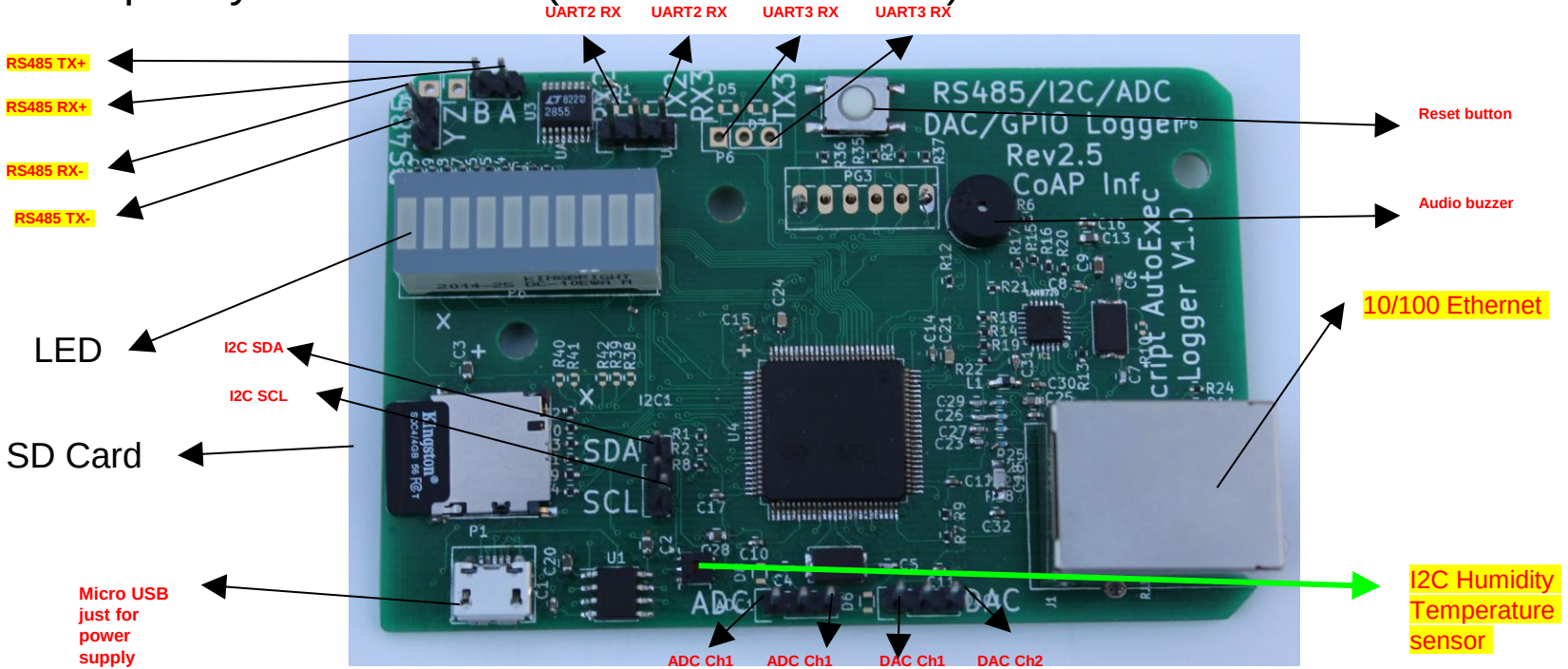




Similar size as
Raspberry PI 2/3

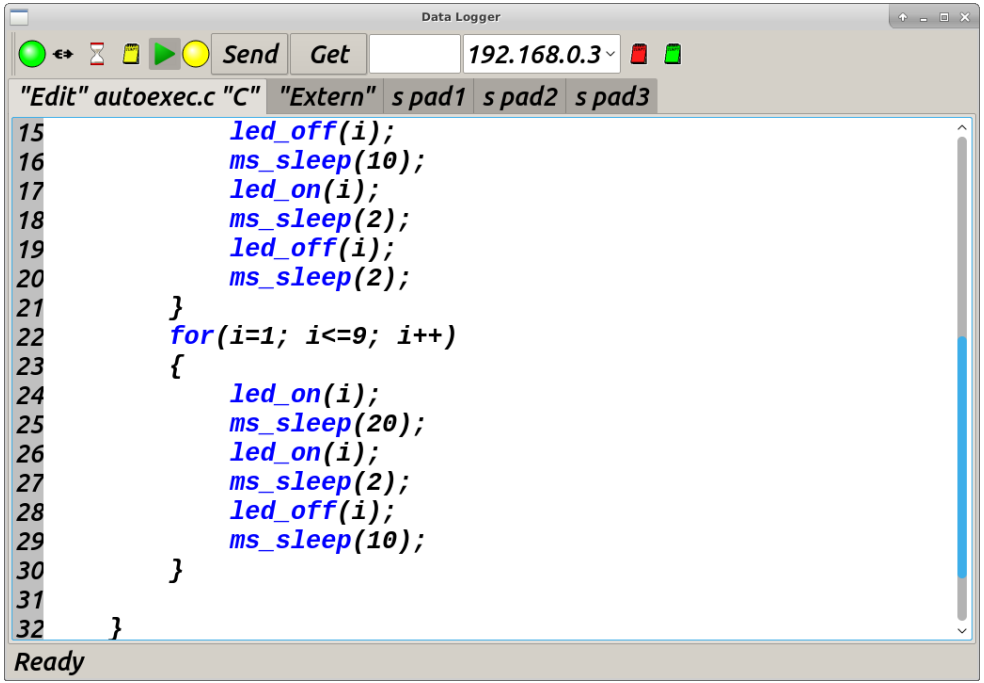
Connector

A. Credit card size board just like raspberry PI, it may be accommodated in raspberry PI enclosure. (Middle PIN Ground)



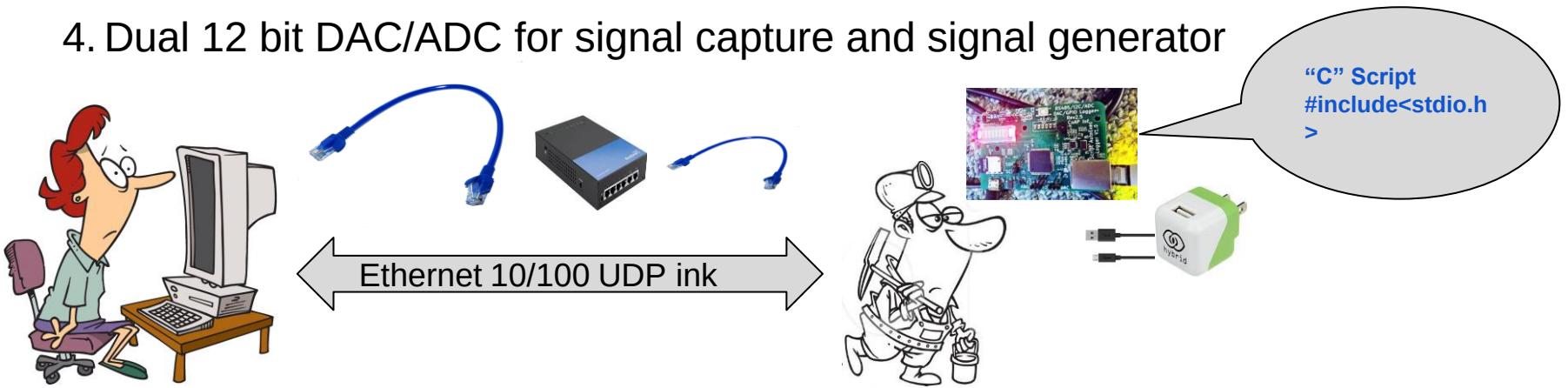
First look into GUI

This utility can download “C” script code into remote IP device with single mouse click. For example program below turns LED on and off.



DIRECT Push script mode

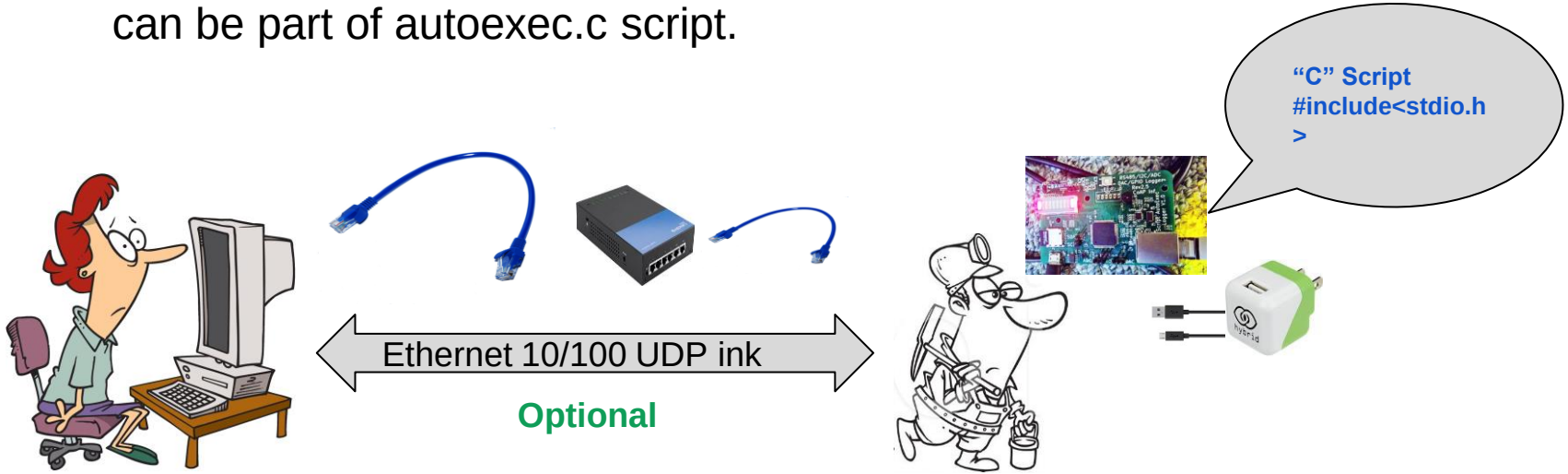
1. User Edit and push “C” script into RAM via CoAP interface.
2. Std user debug printf from “C” script are transported via fast UDP pipe to PC
3. TTL UART/RS485
4. Dual 12 bit DAC/ADC for signal capture and signal generator



Autonomous Mode

1. System checks if “**AUTORUN=1**” in file **autoexec.ini** stored in SD Card, “C” script “**autoexec.c**” starts running as soon as device turns on.

2. All captured data being saved/Used to/from into SD Card.Active feed back can be part of autoexec.c script.



Prerequisites

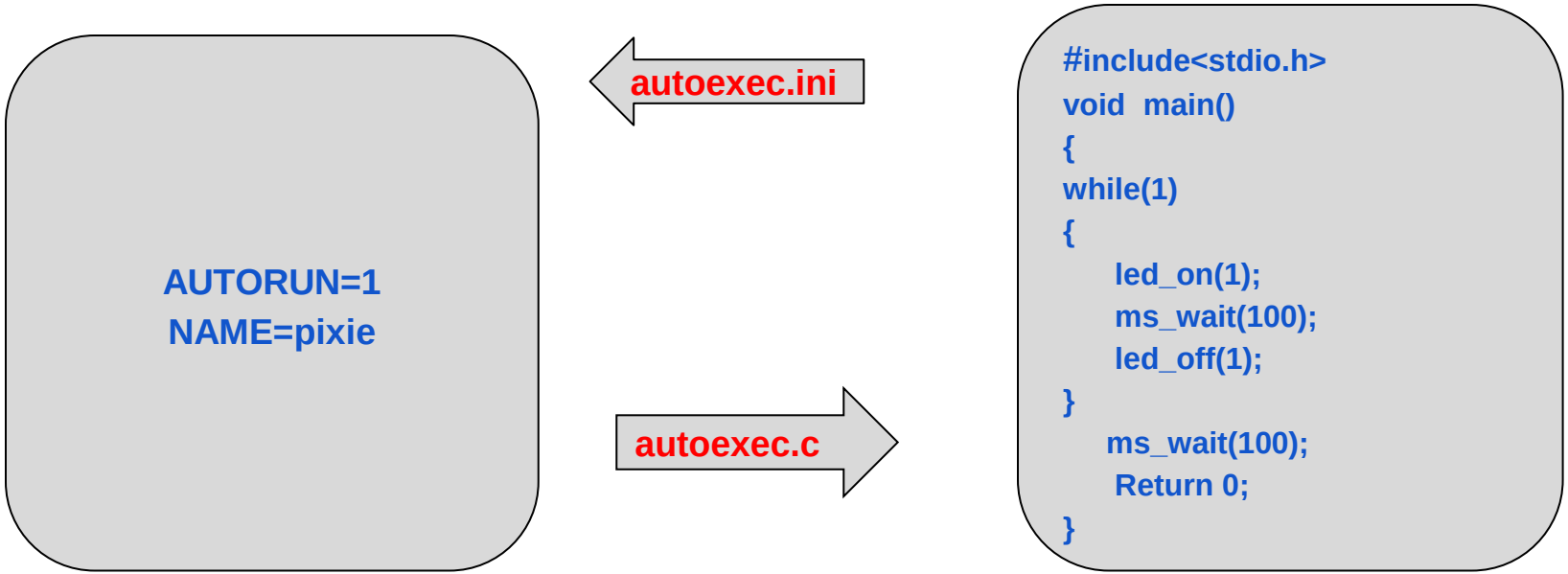
1. Ethernet cable **RJ45**.
2. **Micro USB** power connector (**Typical Android phone charger will work**)
3. **DHCP server** where device can get **iP address**.
4. Network connected Linux/Windows computer with same DHCP server.
5. Few connection cable (some small cables are supplied)

Open box and connect to USB cable, one LED should turn On as soon as power is applied. Connect ethernet cable to router and hardware. Start Linux/Windows computer and launch following GUI application. **Device broadcasts its IP address** once every **5 seconds**. PC application captures broadcast packet and autoconnect

How to I run first autonomous mode data logger

1. Set “AUTORUN=1” in autoexec.ini in root of SD Card.

2. Create autoexec.c file, it will be blinking LED after power ON



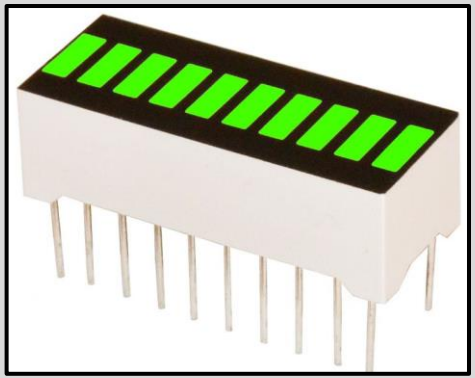
Hello LED (Autonomous mode)

AUTORUN=1
NAME=pixie

autoexec.ini

autoexec.c

```
#include<stdio.h>
void  main()
{
while(1)
{
     led_on(1);
     ms_wait(100);
     led_off(1);
     ms_wait(100);
}
    ms_wait(100);
     Return 0;
}
```



This script will cause first led to be on for 100 millisecond and off for 100 millisecond.

Hello BUZZER (Autonomous mode)

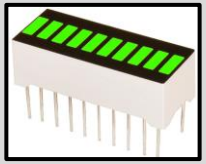
```
AUTORUN=1
NAME=pixie
```

autoexec.ini

autoexec.c

```
#include<stdio.h>
void  main()
{

      led_on(1);
      ms_wait(100);
      led_off(1);
      ms_wait(100);
      beep_ms(1000);
      Return 0;
}
```



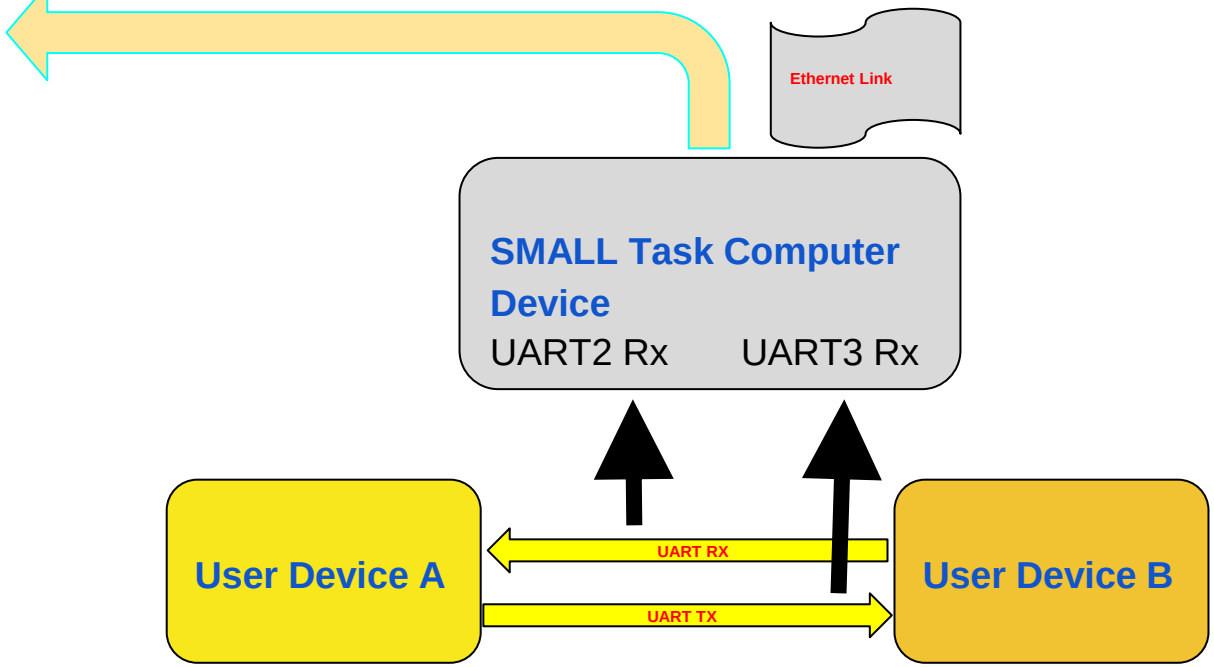
This script will cause first led to be on for 100 millisecond and off for 100 millisecond, followed by a beep sound.

Non Intrusive UART Bidirectional Sniffing/Logging

Single Richtextbox Ouput color (UART2 Red Rx,UART3 Green Rx)

0x19 0x12 0xac
0x31 0xb2 0x9c
(HEX)

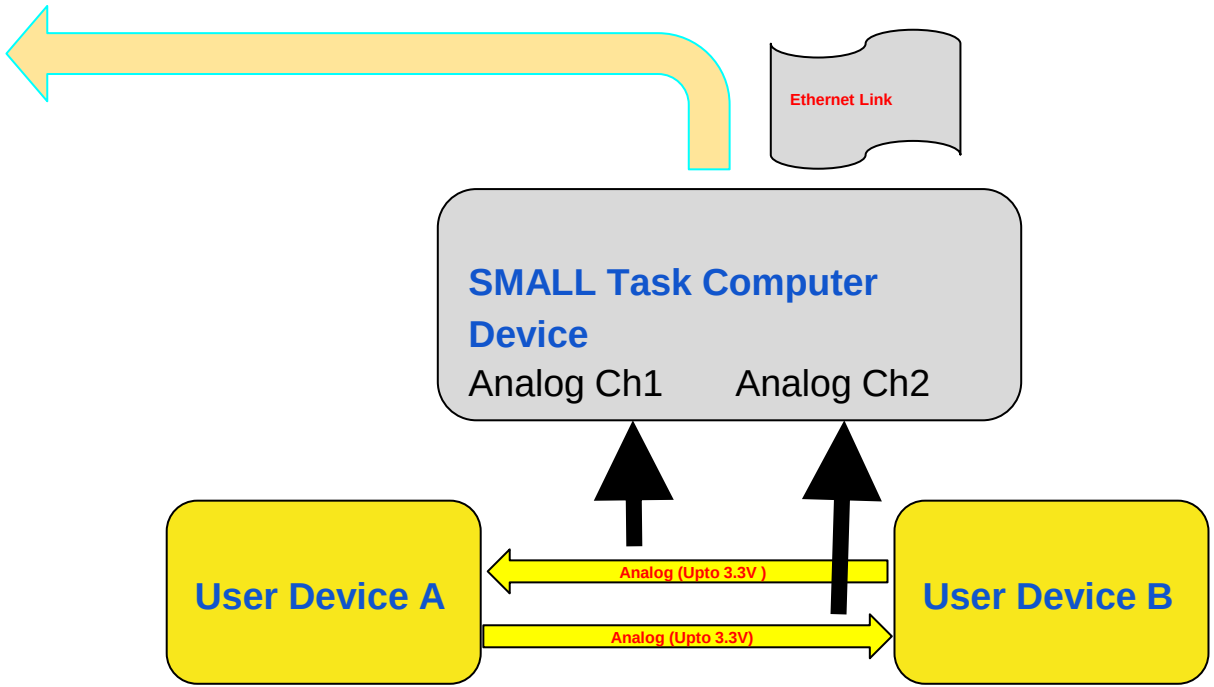
THIS IS TESTFROM UART2 RX
0UART3 RX CONTENT
(ASCII Tex)



Flexible Analog Logging

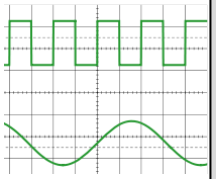
Single Richtextbox Ouput color (Analog 1MHZ Sampling, 12 Bit)

1.0f 1.1f 0.123f
1.30f



Analog Signal Generator

10.5 Mega samples per second
Direct play of signal from SDIO Card to DAC



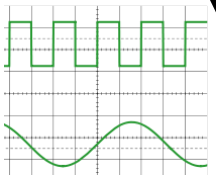
CSV Text File
Ch1,Ch2

1.23,1.1
1.20,1.2
3.10,2.2

Onboard SDIO Card(FAT32)

SMALL Task Computer Device
DAC Ch1 DAC Ch2

Autonomous Start/User Trigger



Analog (Upto 3.3V)

Analog (Upto 3.3V)

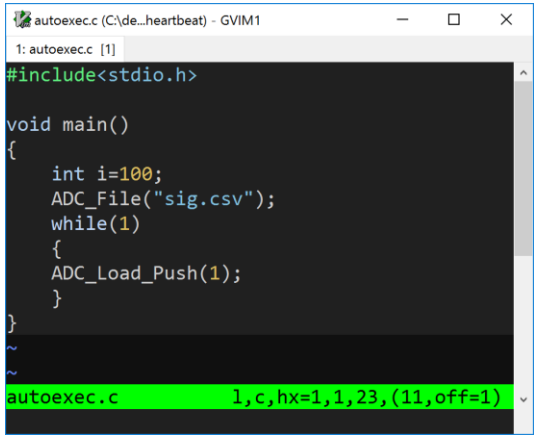
User Device

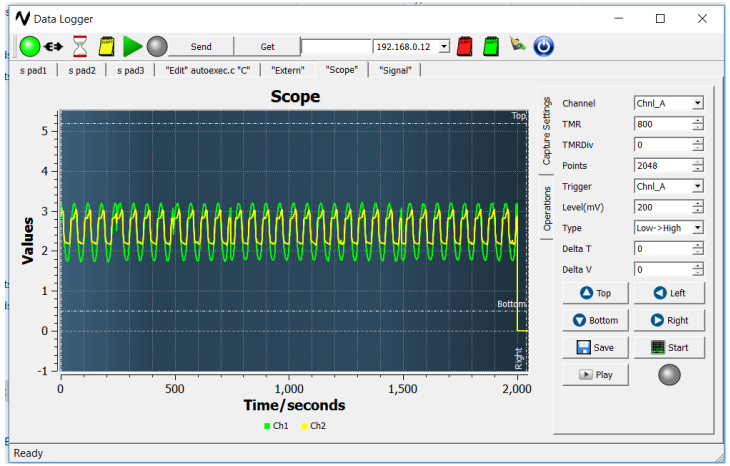
Sample “C” Script diff purpose (Generating signal,CSV)

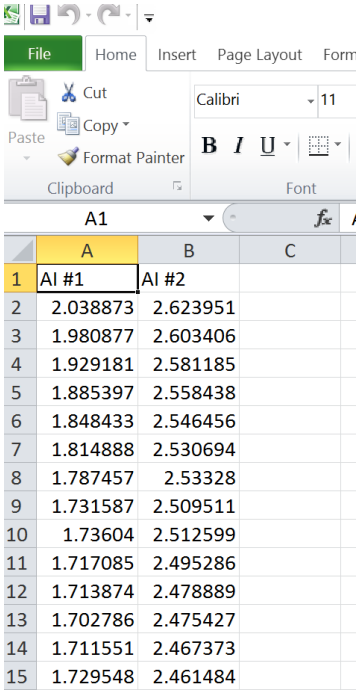
Step 1. Save signal file CSV into root of SDCARD (FAT32).

Step 2. Put following autoexec.c file in SD card,

alongwith sig.csv (SD card size and FAT32 size is max limit)







	A	B	C
1	AI #1	AI #2	
2	2.038873	2.623951	
3	1.980877	2.603406	
4	1.929181	2.581185	
5	1.885397	2.558438	
6	1.848433	2.546456	
7	1.814888	2.530694	
8	1.787457	2.53328	
9	1.731587	2.509511	
10	1.73604	2.512599	
11	1.717085	2.495286	
12	1.713874	2.478889	
13	1.702786	2.475427	
14	1.711551	2.467373	
15	1.729548	2.461484	

Important Note

1. Please note that system is able parse “C” script, but error in “C” syntax can cause script decoder to crash. Please run so PC based error check utility before downloading it in hardware.

2. Small syntax error can cause loss of time spent in debugging so please use tools to check error.

3. Maximum size of script can be 6 KByte, as all complex functions are encapsulated in ROM so it plenty for any purpose.

4. Next Release will have on board SDRAM, which will enable end user with unlimited size of “C” script.

Permanent deployment for purpose of logging

1.Non stop logging of 2UART and 1RS485 Data into SD Card.

2.Humidity / Temperature logging into CSV file .

3.Dual Channel ADC 12bit logging to CSV file.

4.Direct dual analog signal generation based off CSV file stored in SD Card.

5.I2C Master signal.

6. Completely open source cross platform QT application.