

\*\*\*\*\*Draft\*\*\*\*\*

# crypto notes 05/05/20

## Starting with TFTP\_Template

## Testing Electronic Codebook (ECB) & AES Cipher Block Chaining (CBC)

\*\*\*\*\*Draft\*\*\*\*\*

Started with the file from “TFTP\_Template.lpr” to create “test\_crypto.lpr” & “test\_crypto.lpi”  
In addition this needs **uTFTP.pas**, **upker7.sh**, and **cmdstftp**.

Compile the project with **“Run/Compile”** or **“Run/Clean up and Build”**.

The image shows a screenshot of the Visual Studio Code (VS Code) editor interface. The top menu bar includes 'File', 'Edit', 'Search', 'View', 'Source', 'Project', 'Run', 'Tools', 'Window', and 'Help'. Below the menu bar is a toolbar with icons for file operations. The main editor area displays the file 'test\_crypto.lpr' with the following code:

```
105 begin
    . {The following 3 lines are logging to the console
    .   CONSOLE_REGISTER_LOGGING:=true;
    .   LoggingConsoleDeviceAdd(ConsoleDeviceGetDefault);
    .   LoggingDeviceSetDefault(LoggingDeviceFindByType(LOGGING_TYPE_CONSOLE));
110 }
    .
    . {The following 2 lines are logging to a file
    .   LoggingDeviceSetTarget(LoggingDeviceFindByType(LOGGING_TYPE_FILE),'c:\utliblogging.log');
    .   LoggingDeviceSetDefault(LoggingDeviceFindByType(LOGGING_TYPE_FILE)); }
115
    .
    . {Create a console window to show what is happening}
    .   LeftWindow:=ConsoleWindowCreate(ConsoleDeviceGetDefault,CONSOLE_POSITION_LEFT,True);
120
    . {Display a startup message on the console}
    .   ConsoleWindowWriteLn(LeftWindow,'Starting TFTP_Template example');
    .   // wait for IP address and SD Card to be initialised.
    .   WaitForSDDrive;
    .   IPAddress := WaitForIPComplete;
125 {Create and start the HTTP Listener for our web status page}
    .   HTTPListener:=THTTPListener.Create;
    .   HTTPListener.Active:=True;
    .   ConsoleWindowWriteLn (LeftWindow, 'Local Address ' + IPAddress);
    .   SetOnMsg (@Msg);
130 {Register the web status page, the "Thread List" page will allow us to see what is happening in }
    .   WebStatusRegister(HTTPListener,'',True);
    .   CryptoInit;
    .   {Cipher algorithms
    .   CRYPTO_CIPHER_ALG_NONE = 0;
    .   CRYPTO_CIPHER_ALG_AES = 1;
135   CRYPTO_CIPHER_ALG_DES = 2;
    .   CRYPTO_CIPHER_ALG_3DES = 3;
    .   CRYPTO_CIPHER_ALG_RC4 = 4;}
140
    . myAlgorithm:=1;
    . {Cipher algorithm CRYPTO_CIPHER_ALG_AES
142   defined in crypto.pas 0 to 4}
    .   ConsoleWindowWriteLn (LeftWindow, 'Cipher algorithm CRYPTO_CIPHER_ALG_AES ' + intToStr(myAlgorit
    .   //myContext:=
145   {Halt this thread}
    .   ThreadHalt(0);
    . end.
```

The right sidebar shows a 'Compile Project' message with the following hints:

- test\_crypto.lpr(43,2) Note: Local variable "TCP" not used
- test\_crypto.lpr(47,2) Note: Local variable "myContext" not used
- test\_crypto.lpr(49,2) Note: Local variable "myKeySize" not used

The status bar at the bottom shows the file path: 'INS /home/devel/Ultibo Projects/test\_crypto/RPI2/test\_crypto.lpr'.

Once the Green bar is displayed it can be transfer to the Ultibo System.

## AESDecryptBlock (128bit)

## Electronic Codebook (ECB)

## AESDecryptBlock (192bit)

## Electronic Codebook (ECB)

## AESDecryptBlock (256bit)

## Electronic Codebook (ECB)

## AESDecryptBlock (128bit)

## Electronic Codebook (ECB)

**AESDecryptBlock (192bit)  
Electronic Codebook (ECB)**

**AESDecryptBlock (256bit)  
Electronic Codebook (ECB)**

After adding APICrypto.pas

In test\_crypto.lpt in

**var**

**AESECBKey:PByte;  
AESECBData:PByte;  
AESECBKey:TAESKey;**

**AESCBKey:PByte;  
AESCBData:PByte;  
AESCBVector:PByte;**

**Cipher:PCipherContext;**

**key:String;  
Data:String;  
Actual:String;  
PData:PString;  
Datalen:LongWord;**

**InKey:LongWord;  
InKeyStr:String;  
InDataStr:String;  
EncryptDecrypt:LongWord;**

With the addition of function below matches APICrypto.pas

```
tstencryption(InKeyStr,InDataStr:String;InKey,EncryptDecrypt:LongWord):String;
var
  AESECBKey:PByte;
  AESECBData:PByte;
  AESECBAESKey:TAESKey;
begin

  AESECBData:=AllocMem(AES_BLOCK_SIZE);
  if(InKey=0) then
    begin
      AESECBKey:=AllocMem(AES_KEY_SIZE128);
      StringToBytes(InKeyStr,PByte(AESECBKey),AES_KEY_SIZE128);
      StringToBytes(InDataStr,PByte(AESECBData),AES_BLOCK_SIZE);
      AESKeySetup(AESECBKey,AES_KEY_SIZE128,@AESECBAESKey);
    end;
  if(InKey=1) then
    begin
      AESECBKey:=AllocMem(AES_KEY_SIZE192);
      StringToBytes(InKeyStr,PByte(AESECBKey),AES_KEY_SIZE192);
      StringToBytes(InDataStr,PByte(AESECBData),AES_BLOCK_SIZE);
      AESKeySetup(AESECBKey,AES_KEY_SIZE192,@AESECBAESKey);
    end;
  if(InKey=2) then
    begin
      AESECBKey:=AllocMem(AES_KEY_SIZE256);
      StringToBytes(InKeyStr,PByte(AESECBKey),AES_KEY_SIZE256);
      StringToBytes(InDataStr,PByte(AESECBData),AES_BLOCK_SIZE);
      AESKeySetup(AESECBKey,AES_KEY_SIZE256,@AESECBAESKey);
    end;

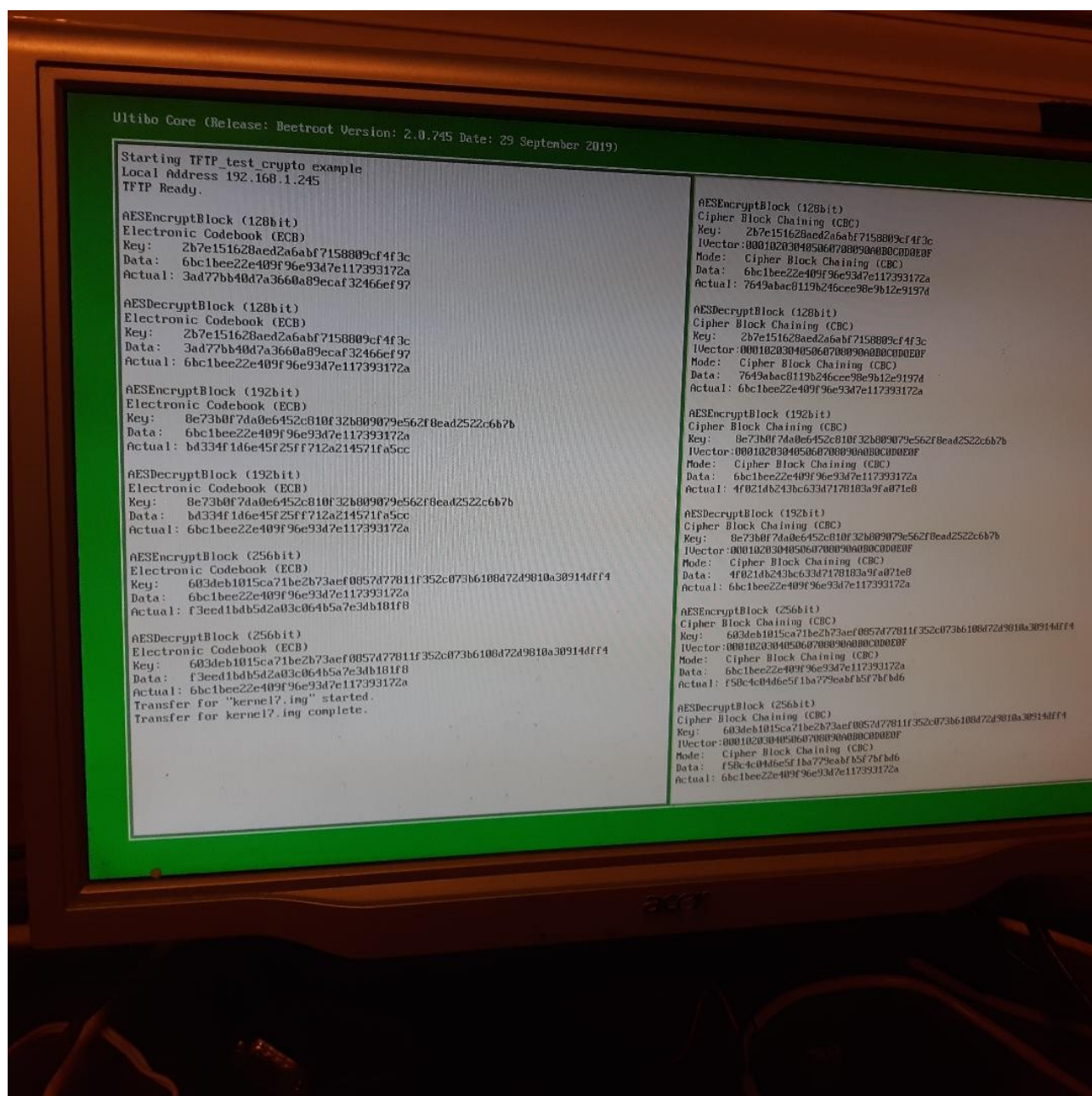
  //AESECBData:=AllocMem(AES_BLOCK_SIZE);

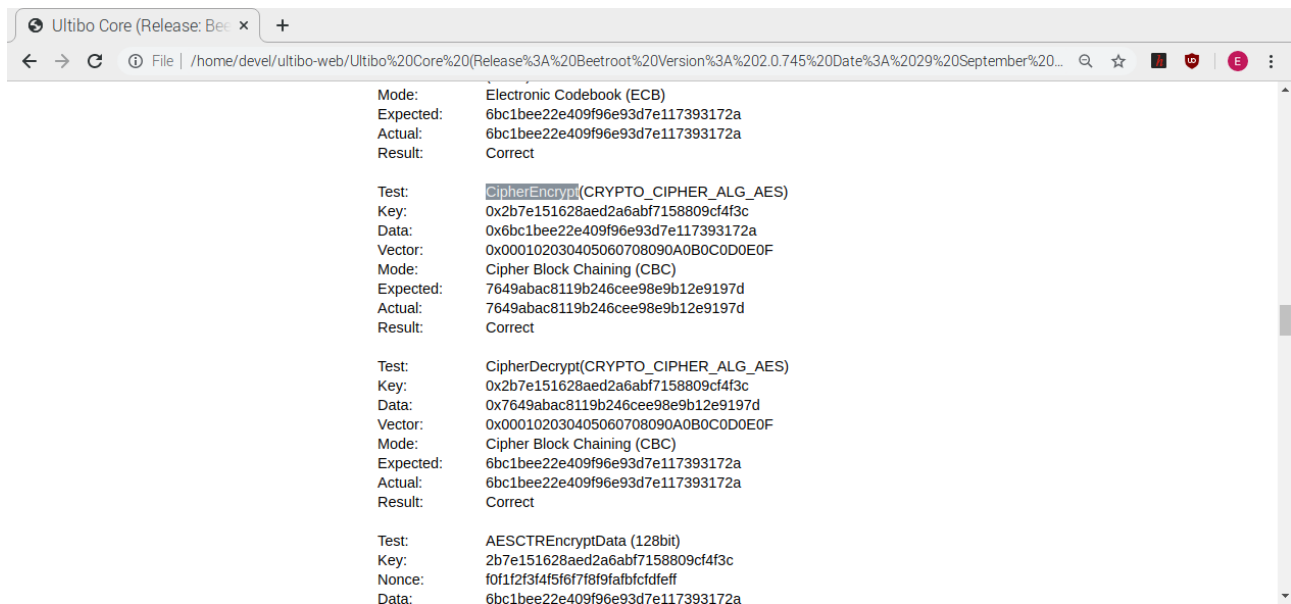
  if(EncryptDecrypt=1) then
    begin
      AESEncryptBlock(AESECBData,AESECBData,@AESECBAESKey);
    end;

  if(EncryptDecrypt=0) then
    begin
      AESDecryptBlock(AESECBData,AESECBData,@AESECBAESKey);
```

end;

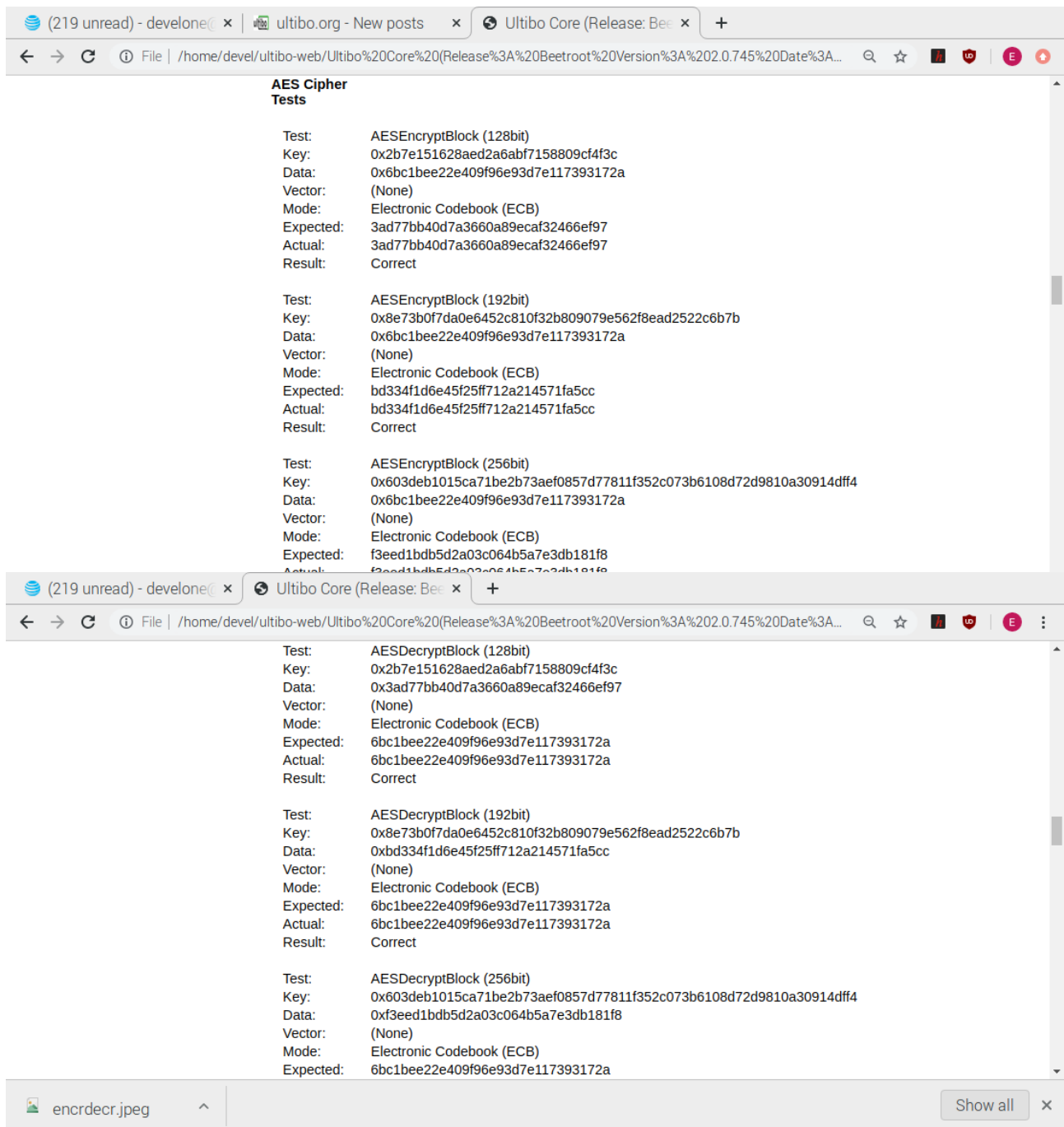
./upker.sh





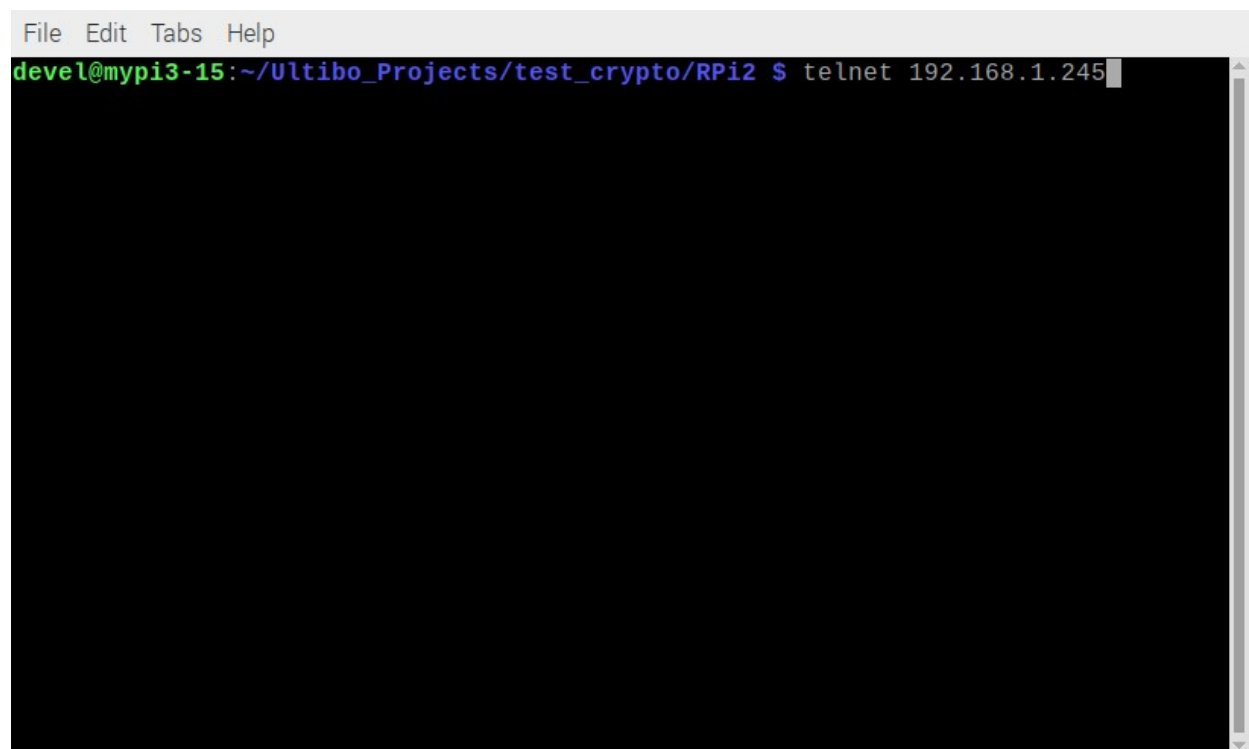
Now the results match the results on <http://192.168.1.245/status/cryptoapi/>





Decryption APICrypto.pas

shell1

A terminal window with a light gray title bar containing the menu items 'File', 'Edit', 'Tabs', and 'Help'. The terminal has a black background. The prompt 'devel@mypi3-15:~/Ultibo\_Projects/test\_crypto/RPi2 \$' is shown in green and blue text. The command 'telnet 192.168.1.245' is entered in blue text, followed by a white cursor. A vertical scrollbar is visible on the right side of the terminal area.

```
File Edit Tabs Help
devel@mypi3-15:~/Ultibo_Projects/test_crypto/RPi2 $ telnet 192.168.1.245
```

shell2



```
File Edit Tabs Help
29-3-20 02:24:18      3798568 start_x.elf
29-3-20 02:24:18      3145850 t
29-3-20 02:24:20      635016 teapot.obj.dat
29-3-20 02:23:56        24 testfile
29-3-20 02:24:20     27983872 test.h264
29-3-20 02:24:24        500 test.html
10-4-20 16:23:58       7848 test.j2k
6-4-20 17:37:26     196730 test_wr.bmp
29-3-20 02:24:24        1718 ultibologging.log
29-3-20 02:24:24     27983872 v1.h264
29-3-20 02:24:30     1002763 v2.h264
29-3-20 02:24:30      <DIR> www
2-4-20 17:31:26      65596 red.pgm
2-4-20 17:31:38      65596 grn.pgm
2-4-20 17:31:52      65596 blu.pgm
6-4-20 11:23:30       1024 Sred.bin
6-4-20 11:23:34       1024 Sgrn.bin
6-4-20 11:23:36     262144 rcgrn.bin
6-4-20 11:23:38       1024 Sblu.bin
6-4-20 11:23:38     262144 rcblu.bin
      69 file(s) 136527430 bytes
      2 dir(s)

C:\>
```

## Webstatus

(211 unread) - deve x | Wifi - Page 2 - ultib x | w Common Vulnerab x | develone/tiny-AES x | Ultibo Core (Releas x +

← → ↻ ⓘ Not secure | 192.168.1.245/status ☆ 🔒 🔐 ⓘ

Ultibo Core (Release: Beetroot Version: 2.0.745 Date: 29 September 2019)

General	General	
Platform	Release Name:	Beetroot
Memory	Release Version:	2.0.745
Heap Blocks	Release Date:	29 September 2019
CPU	Time (Local):	30-12-99 00:00:08
FPU	Time (UTC):	30-12-99 00:00:08
GPU	Timezone:	UTC
RTL	Daylight Start:	None
Clock	Daylight Date:	N/A
Locale	Standard Start:	None
Threading	Standard Date:	N/A
Thread List	Uptime:	0 days 00:00:08
Scheduler		
Devices		
Drivers		
Handles		
USB		
MMC / SD		
Network		
Storage		
Filesystem		
Disk Cache		
Keyboard		
Mouse		
Framebuffer		