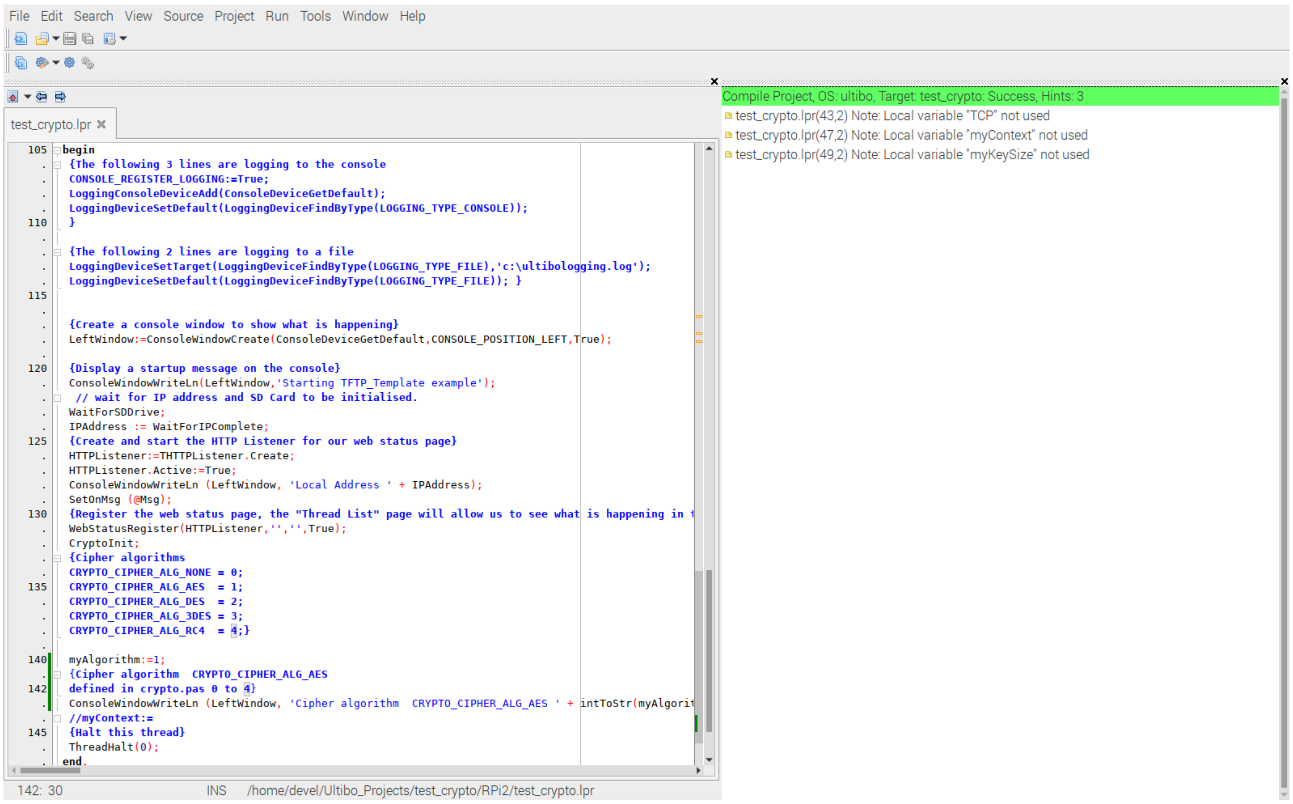
***********Draft***********

crypto notes 05/07/20
Starting with TFTP_Template
Testing Electronic Codebook (ECB) & AES Cipher Block Chaining (CBC)

***********Draft***********

Started with the file from **“TFTP_Template.lpr”** to creare **“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”.**



```
begin
 {The following 3 lines are logging to the console
 CONSOLE_REGISTER_LOGGING:=True;
 LoggingConsoleDeviceAdd(ConsoleDeviceGetDefault);
 LoggingDeviceSetDefault(LoggingDeviceFindByType(LOGGING_TYPE_CONSOLE));
 }

 {The following 2 lines are logging to a file
 LoggingDeviceSetTarget(LoggingDeviceFindByType(LOGGING_TYPE_FILE),'c:\ultibologging.log');
 LoggingDeviceSetDefault(LoggingDeviceFindByType(LOGGING_TYPE_FILE)); }


 {Create a console window to show what is happening}
 LeftWindow:=ConsoleWindowCreate(ConsoleDeviceGetDefault,CONSOLE_POSITION_LEFT,True);

 {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;
 {Create and start the HTTP Listener for our web status page}
 HTTPListener:=THTTPListener.Create;
 HTTPListener.Active:=True;
 ConsoleWindowWriteLn (LeftWindow, 'Local Address ' + IPAddress);
 SetOnMsg (@Msg);
 {Register the web status page, the "Thread List" page will allow us to see what is happening in t
 WebStatusRegister(HTTPListener,'','',True);
 CryptoInit;
 {Cipher algorithms
 CRYPTO_CIPHER_ALG_NONE = 0;
 CRYPTO_CIPHER_ALG_AES  = 1;
 CRYPTO_CIPHER_ALG_DES  = 2;
 CRYPTO_CIPHER_ALG_3DES = 3;
 CRYPTO_CIPHER_ALG_RC4  = 4;}

 myAlgorithm:=1;
 {Cipher algorithm  CRYPTO_CIPHER_ALG_AES
 defined in crypto.pas 0 to 4}
 ConsoleWindowWriteLn (LeftWindow, 'Cipher algorithm  CRYPTO_CIPHER_ALG_AES ' + intToStr(myAlgorit
 //myContext:=
 {Halt this thread}
 ThreadHalt(0);
end.
```

```
Compile Project, OS: ultibo, Target: test_crypto: Success, Hints: 3
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
```

```
142: 30    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.

AESEncryptBlock (128bit)
Electronic Codebook (ECB)

AESEncryptBlock (192bit)
Electronic Codebook (ECB)

AESEncryptBlock (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;
 AESECBAESKey:TAESKey;

 AESCBCKey:PByte;
 AESCBCData:PByte;
 AESCBCVector: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):St
ring;
  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
Testing 2 blocks
The program test_crypto.lpr now has 2 functions in support of encrption/decryption

Electronic Codebook (ECB)

function ecbencryption(InKeyStr,InDataStr:String;InKey,EncryptDecrypt:LongWord):String;

Cipher Block Chaining (CBC)

function cbcencryption(InKeyStr,InDataStr,InIVStr:String;InKey,EncryptDecrypt:LongWord):String;

Steps to encrypt a block of data.

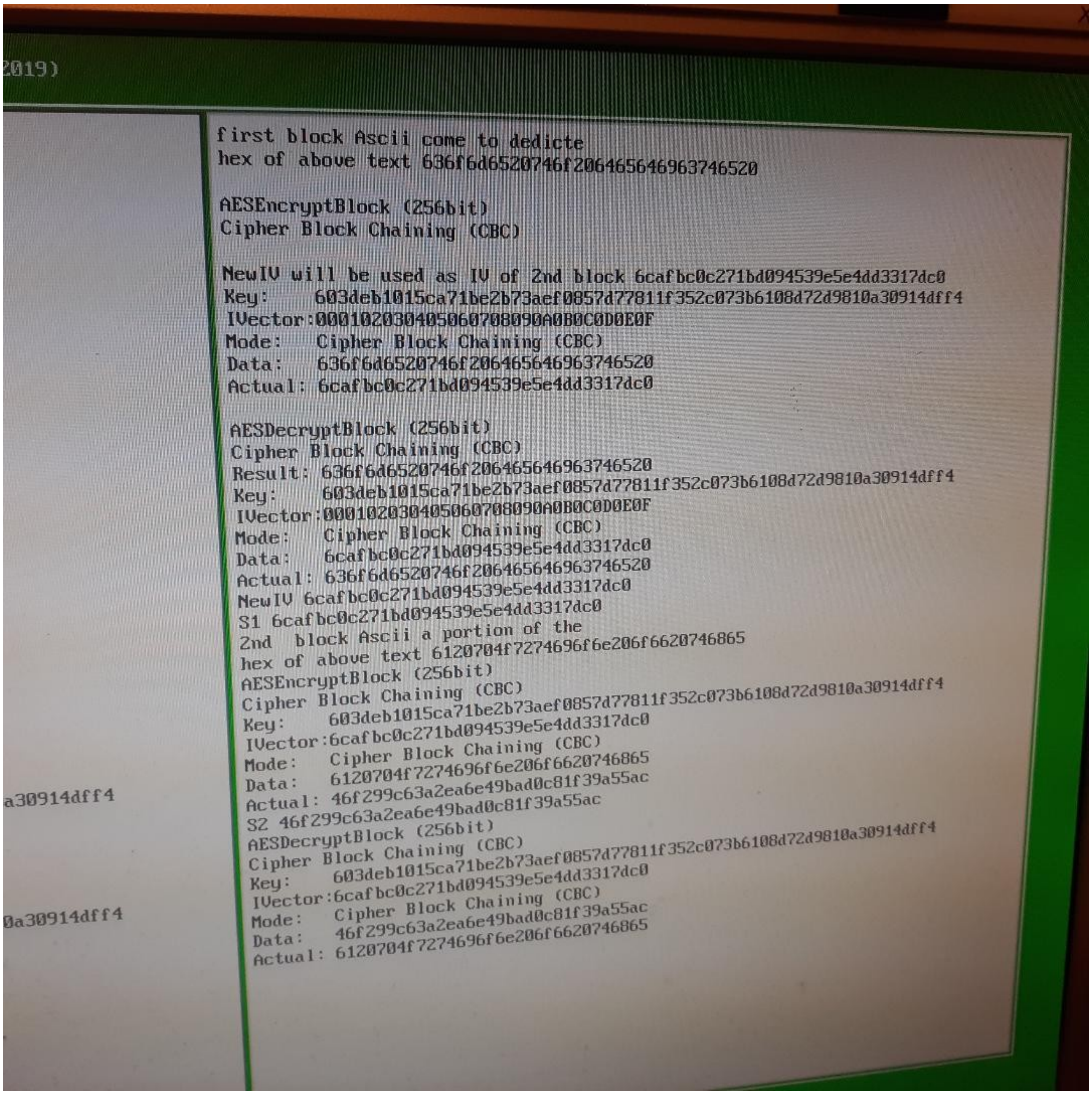
1. Split the data in blocks of 128bits.
   This is what makes up
   Example 16 characters would
   012345678901234567
   'come to dedicte '
   make a block of
   a 128bit block hex when converted from Ascii.
   '636f6d6520746f206465646963746520'

2. Encrypt the first block using a key (128bits, 192bits, or 256bits) using
   the Cipher Block Chaining (CBC) mode and IVector.
   Below are example of (128bits, 192bits, or 256bits)
   128bits
   '2b7e151628aed2a6abf7158809cf4f3c'
   192bits
   '8e73b0f7da0e6452c810f32b809079e562f8ead2522c6b7b'
   256bits
   '603deb1015ca71be2b73aef0857d77811f352c073b6108d72d9810a30914dff4'
   Below is an example IVector
   '000102030405060708090A0B0C0D0E0F'
   The result of the first block will be used as the IVector for the 2nd block.
   With the 256bits as key, the function cbcencryption was used to encryt 2 blocks
   Key '603deb1015ca71be2b73aef0857d77811f352c073b6108d72d9810a30914dff4'
   IVector '000102030405060708090A0B0C0D0E0F'
   Data '636f6d6520746f206465646963746520'

   NewIV for 2nd block '6cafbc0c271b094529e54dd2217dc0'

3. Note: Step3 is optional Decrypt the first block using the same size key to verify that everthing is working okay.
   The same IVector needs to be used.

4. The result of the first block will be used as the IVector for the 2nd block.

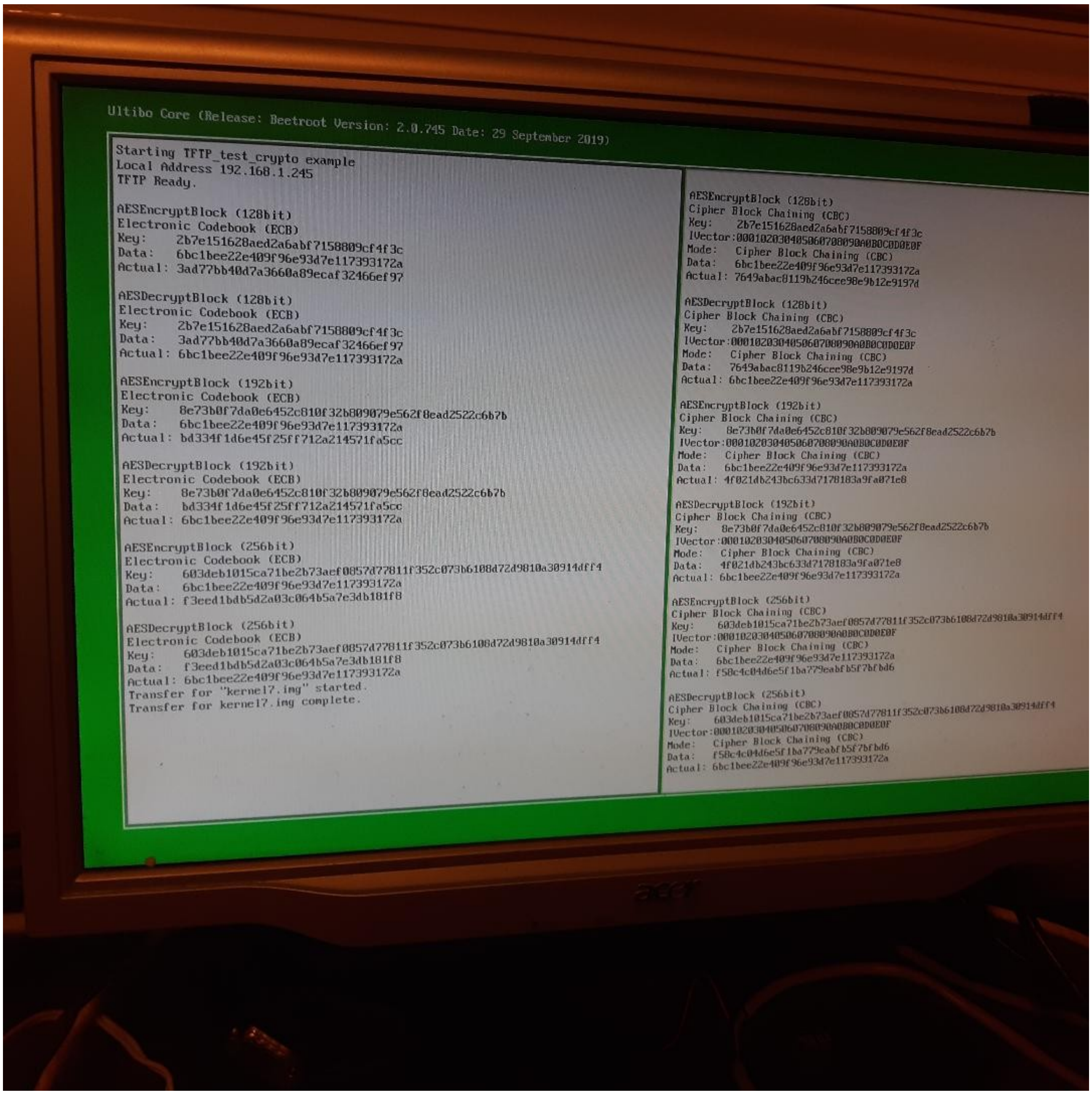


```
2019)
```

```
first block Ascii come to dedicte
hex of above text 636f6d6520746f20646564696374652O

AESEncryptBlock (256bit)
Cipher Block Chaining (CBC)

NewIV will be used as IV of 2nd block 6cafbc0c271bd094539e5e4dd3317dc0
Key:     603deb1015ca71be2b73aef0857d77811f352c073b6108d72d9810a30914dff4
IVector:000102030405060708090A0B0C0D0E0F
Mode:    Cipher Block Chaining (CBC)
Data:    636f6d6520746f206465646963746520
Actual:  6cafbc0c271bd094539e5e4dd3317dc0

AESDecryptBlock (256bit)
Cipher Block Chaining (CBC)
Result:  636f6d6520746f206465646963746520
Key:     603deb1015ca71be2b73aef0857d77811f352c073b6108d72d9810a30914dff4
IVector:000102030405060708090A0B0C0D0E0F
Mode:    Cipher Block Chaining (CBC)
Data:    6cafbc0c271bd094539e5e4dd3317dc0
Actual:  636f6d6520746f206465646963746520
NewIV 6cafbc0c271bd094539e5e4dd3317dc0
S1 6cafbc0c271bd094539e5e4dd3317dc0
2nd  block Ascii a portion of the
hex of above text 6120704f7274696f6e206f6620746865
AESEncryptBlock (256bit)
Cipher Block Chaining (CBC)
Key:     603deb1015ca71be2b73aef0857d77811f352c073b6108d72d9810a30914dff4
IVector:6cafbc0c271bd094539e5e4dd3317dc0
Mode:    Cipher Block Chaining (CBC)
Data:    6120704f7274696f6e206f6620746865
Actual:  46f299c63a2ea6e49bad0c81f39a55ac
S2 46f299c63a2ea6e49bad0c81f39a55ac
AESDecryptBlock (256bit)
Cipher Block Chaining (CBC)
Key:     603deb1015ca71be2b73aef0857d77811f352c073b6108d72d9810a30914dff4
IVector:6cafbc0c271bd094539e5e4dd3317dc0
Mode:    Cipher Block Chaining (CBC)
Data:    46f299c63a2ea6e49bad0c81f39a55ac
Actual:  6120704f7274696f6e206f6620746865
```

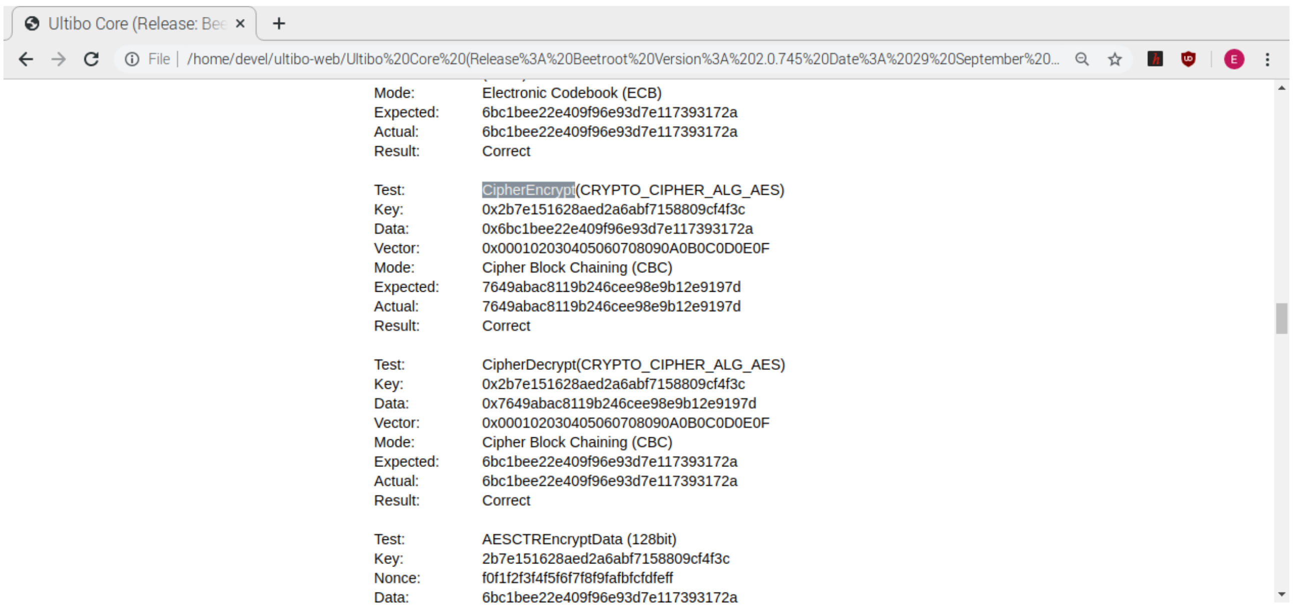
```
a30914dff4
0a30914dff4
```

Testing of ECB & CBC



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

Starting TFTP_test_crypto example
Local Address 192.168.1.245
TFTP Ready.

AESEncryptBlock (128bit)
Electronic Codebook (ECB)
Key:    2b7e151628aed2a6abf7158809cf4f3c
Data:   6bc1bee22e409f96e93d7e117393172a
Actual: 3ad77bb40d7a3660a89ecaf32466ef97

AESDecryptBlock (128bit)
Electronic Codebook (ECB)
Key:    2b7e151628aed2a6abf7158809cf4f3c
Data:   3ad77bb40d7a3660a89ecaf32466ef97
Actual: 6bc1bee22e409f96e93d7e117393172a

AESEncryptBlock (192bit)
Electronic Codebook (ECB)
Key:    8e73b0f7da0e6452c810f32b809079e562f8ead2522c6b7b
Data:   6bc1bee22e409f96e93d7e117393172a
Actual: bd334f1d6e45f25ff712a214571fa5cc

AESDecryptBlock (192bit)
Electronic Codebook (ECB)
Key:    8e73b0f7da0e6452c810f32b809079e562f8ead2522c6b7b
Data:   bd334f1d6e45f25ff712a214571fa5cc
Actual: 6bc1bee22e409f96e93d7e117393172a

AESEncryptBlock (256bit)
Electronic Codebook (ECB)
Key:    603deb1015ca71be2b73aef0857d77811f352c073b6108d72d9810a30914dff4
Data:   6bc1bee22e409f96e93d7e117393172a
Actual: f3eed1bdb5d2a03c064b5a7e3db181f8

AESDecryptBlock (256bit)
Electronic Codebook (ECB)
Key:    603deb1015ca71be2b73aef0857d77811f352c073b6108d72d9810a30914dff4
Data:   f3eed1bdb5d2a03c064b5a7e3db181f8
Actual: 6bc1bee22e409f96e93d7e117393172a
Transfer for "kernel7.img" started.
Transfer for kernel7.img complete.
```

```
AESEncryptBlock (128bit)
Cipher Block Chaining (CBC)
Key:    2b7e151628aed2a6abf7158809cf4f3c
IVector:000102030405060708090A0B0C0D0E0F
Mode:   Cipher Block Chaining (CBC)
Data:   6bc1bee22e409f96e93d7e117393172a
Actual: 7649abac8119b246cee98e9b12e9197d

AESDecryptBlock (128bit)
Cipher Block Chaining (CBC)
Key:    2b7e151628aed2a6abf7158809cf4f3c
IVector:000102030405060708090A0B0C0D0E0F
Mode:   Cipher Block Chaining (CBC)
Data:   7649abac8119b246cee98e9b12e9197d
Actual: 6bc1bee22e409f96e93d7e117393172a

AESEncryptBlock (192bit)
Cipher Block Chaining (CBC)
Key:    8e73b0f7da0e6452c810f32b809079e562f8ead2522c6b7b
IVector:000102030405060708090A0B0C0D0E0F
Mode:   Cipher Block Chaining (CBC)
Data:   6bc1bee22e409f96e93d7e117393172a
Actual: 4f021db243bc633d7178183a9fa071e8

AESDecryptBlock (192bit)
Cipher Block Chaining (CBC)
Key:    8e73b0f7da0e6452c810f32b809079e562f8ead2522c6b7b
IVector:000102030405060708090A0B0C0D0E0F
Mode:   Cipher Block Chaining (CBC)
Data:   4f021db243bc633d7178183a9fa071e8
Actual: 6bc1bee22e409f96e93d7e117393172a

AESEncryptBlock (256bit)
Cipher Block Chaining (CBC)
Key:    603deb1015ca71be2b73aef0857d77811f352c073b6108d72d9810a30914dff4
IVector:000102030405060708090A0B0C0D0E0F
Mode:   Cipher Block Chaining (CBC)
Data:   6bc1bee22e409f96e93d7e117393172a
Actual: f58c4c04d6e5f1ba779eabfb5f7bfbd6

AESDecryptBlock (256bit)
Cipher Block Chaining (CBC)
Key:    603deb1015ca71be2b73aef0857d77811f352c073b6108d72d9810a30914dff4
IVector:000102030405060708090A0B0C0D0E0F
Mode:   Cipher Block Chaining (CBC)
Data:   f58c4c04d6e5f1ba779eabfb5f7bfbd6
Actual: 6bc1bee22e409f96e93d7e117393172a
```

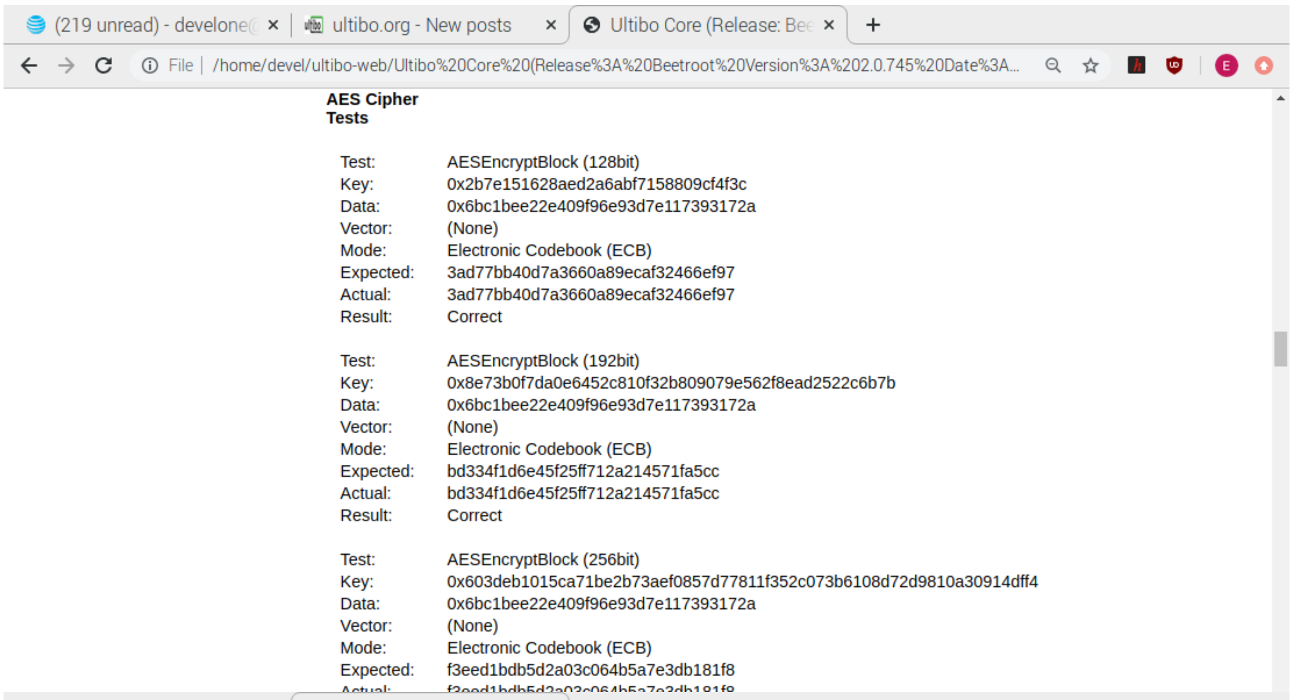


Ultibo Core (Release: Bee

File | /home/devel/ultibo-web/Ultibo%20Core%20(Release%3A%20Beetroot%20Version%3A%202.0.745%20Date%3A%2029%20September%20...

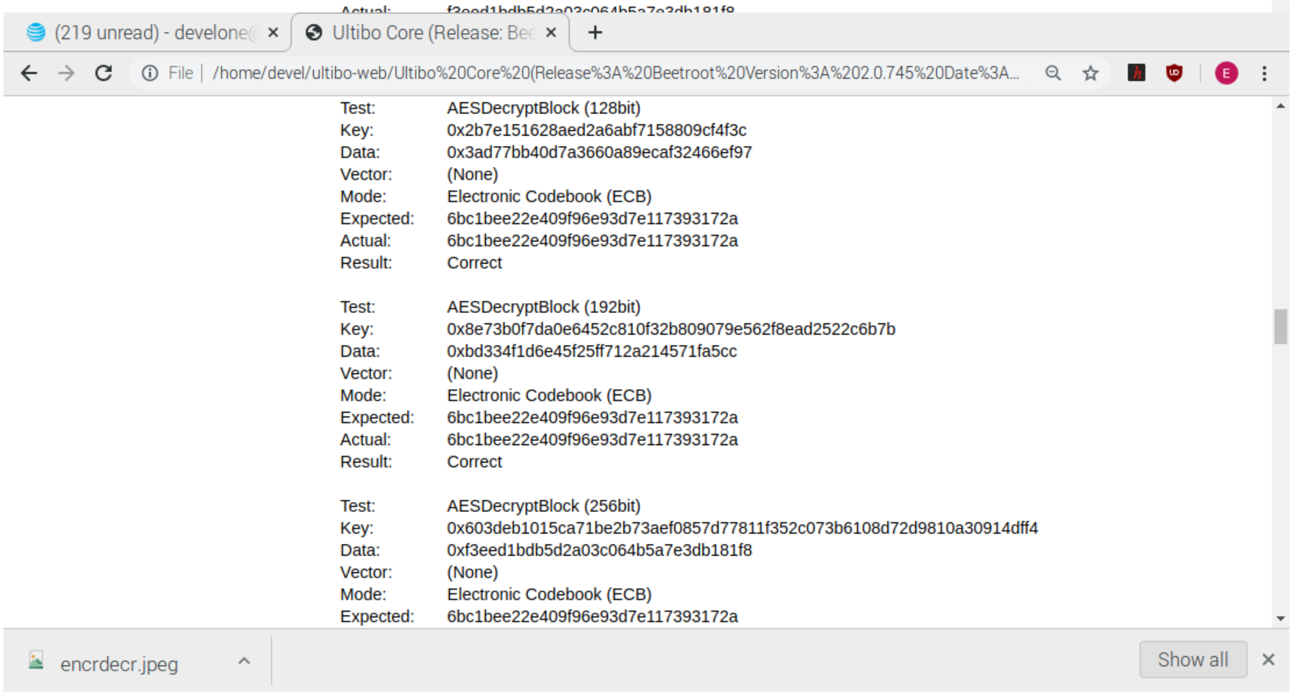
```
Mode:        Electronic Codebook (ECB)
Expected:    6bc1bee22e409f96e93d7e117393172a
Actual:      6bc1bee22e409f96e93d7e117393172a
Result:      Correct

Test:        CipherEncrypt(CRYPTO_CIPHER_ALG_AES)
Key:         0x2b7e151628aed2a6abf7158809cf4f3c
Data:        0x6bc1bee22e409f96e93d7e117393172a
Vector:      0x000102030405060708090A0B0C0D0E0F
Mode:        Cipher Block Chaining (CBC)
Expected:    7649abac8119b246cee98e9b12e9197d
Actual:      7649abac8119b246cee98e9b12e9197d
Result:      Correct

Test:        CipherDecrypt(CRYPTO_CIPHER_ALG_AES)
Key:         0x2b7e151628aed2a6abf7158809cf4f3c
Data:        0x7649abac8119b246cee98e9b12e9197d
Vector:      0x000102030405060708090A0B0C0D0E0F
Mode:        Cipher Block Chaining (CBC)
Expected:    6bc1bee22e409f96e93d7e117393172a
Actual:      6bc1bee22e409f96e93d7e117393172a
Result:      Correct

Test:        AESCTREncryptData (128bit)
Key:         2b7e151628aed2a6abf7158809cf4f3c
Nonce:       f0f1f2f3f4f5f6f7f8f9fafbfcfdfeff
Data:        6bc1bee22e409f96e93d7e117393172a
```

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


(219 unread) - develone
ultibo.org - New posts
Ultibo Core (Release: Bee
File | /home/devel/ultibo-web/Ultibo%20Core%20(Release%3A%20Beetroot%20Version%3A%202.0.745%20Date%3A...

AES Cipher Tests

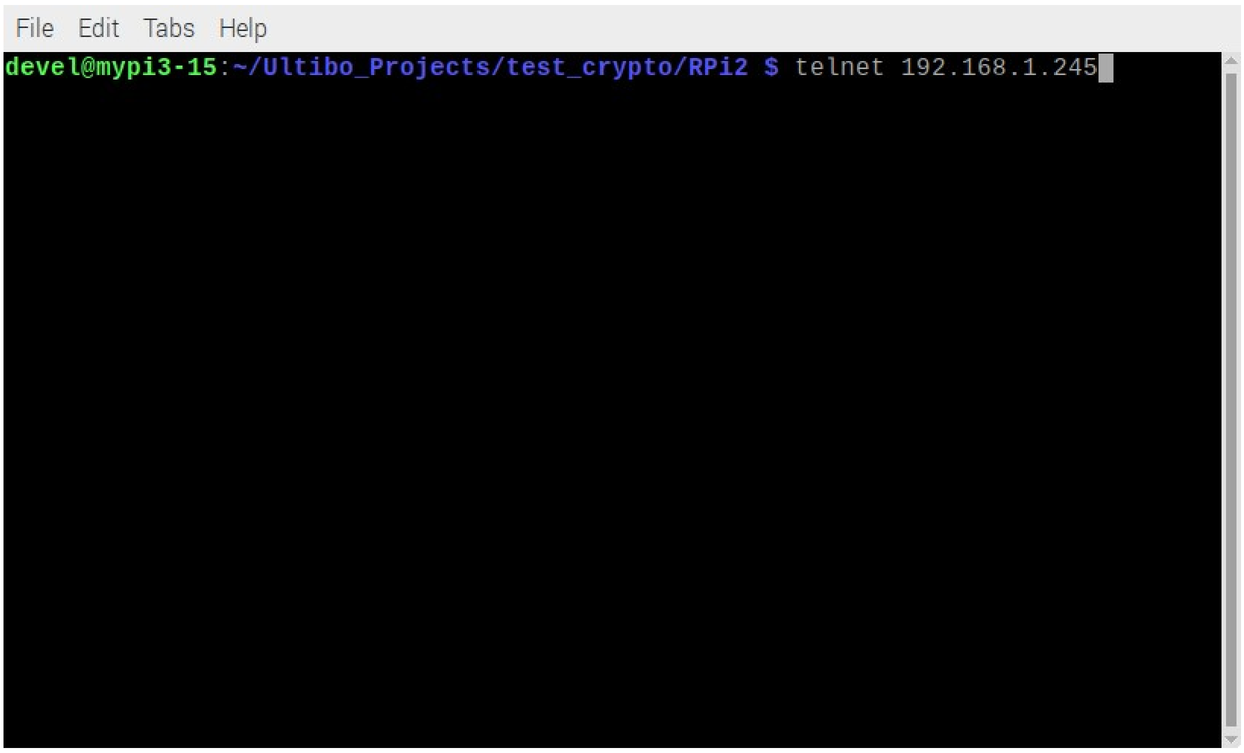
```
Test:       AESEncryptBlock (128bit)
Key:        0x2b7e151628aed2a6abf7158809cf4f3c
Data:       0x6bc1bee22e409f96e93d7e117393172a
Vector:     (None)
Mode:       Electronic Codebook (ECB)
Expected:   3ad77bb40d7a3660a89ecaf32466ef97
Actual:     3ad77bb40d7a3660a89ecaf32466ef97
Result:     Correct

Test:       AESEncryptBlock (192bit)
Key:        0x8e73b0f7da0e6452c810f32b809079e562f8ead2522c6b7b
Data:       0x6bc1bee22e409f96e93d7e117393172a
Vector:     (None)
Mode:       Electronic Codebook (ECB)
Expected:   bd334f1d6e45f25ff712a214571fa5cc
Actual:     bd334f1d6e45f25ff712a214571fa5cc
Result:     Correct

Test:       AESEncryptBlock (256bit)
Key:        0x603deb1015ca71be2b73aef0857d77811f352c073b6108d72d9810a30914dff4
Data:       0x6bc1bee22e409f96e93d7e117393172a
Vector:     (None)
Mode:       Electronic Codebook (ECB)
Expected:   f3eed1bdb5d2a03c064b5a7e3db181f8
Actual:     f3eed1bdb5d2a03c064b5a7e3db181f8
```


(219 unread) - develone
Ultibo Core (Release: Bee
File | /home/devel/ultibo-web/Ultibo%20Core%20(Release%3A%20Beetroot%20Version%3A%202.0.745%20Date%3A...
encrdecr.jpeg
Show all

```
Test:       AESDecryptBlock (128bit)
Key:        0x2b7e151628aed2a6abf7158809cf4f3c
Data:       0x3ad77bb40d7a3660a89ecaf32466ef97
Vector:     (None)
Mode:       Electronic Codebook (ECB)
Expected:   6bc1bee22e409f96e93d7e117393172a
Actual:     6bc1bee22e409f96e93d7e117393172a
Result:     Correct

Test:       AESDecryptBlock (192bit)
Key:        0x8e73b0f7da0e6452c810f32b809079e562f8ead2522c6b7b
Data:       0xbd334f1d6e45f25ff712a214571fa5cc
Vector:     (None)
Mode:       Electronic Codebook (ECB)
Expected:   6bc1bee22e409f96e93d7e117393172a
Actual:     6bc1bee22e409f96e93d7e117393172a
Result:     Correct

Test:       AESDecryptBlock (256bit)
Key:        0x603deb1015ca71be2b73aef0857d77811f352c073b6108d72d9810a30914dff4
Data:       0xf3eed1bdb5d2a03c064b5a7e3db181f8
Vector:     (None)
Mode:       Electronic Codebook (ECB)
Expected:   6bc1bee22e409f96e93d7e117393172a
```

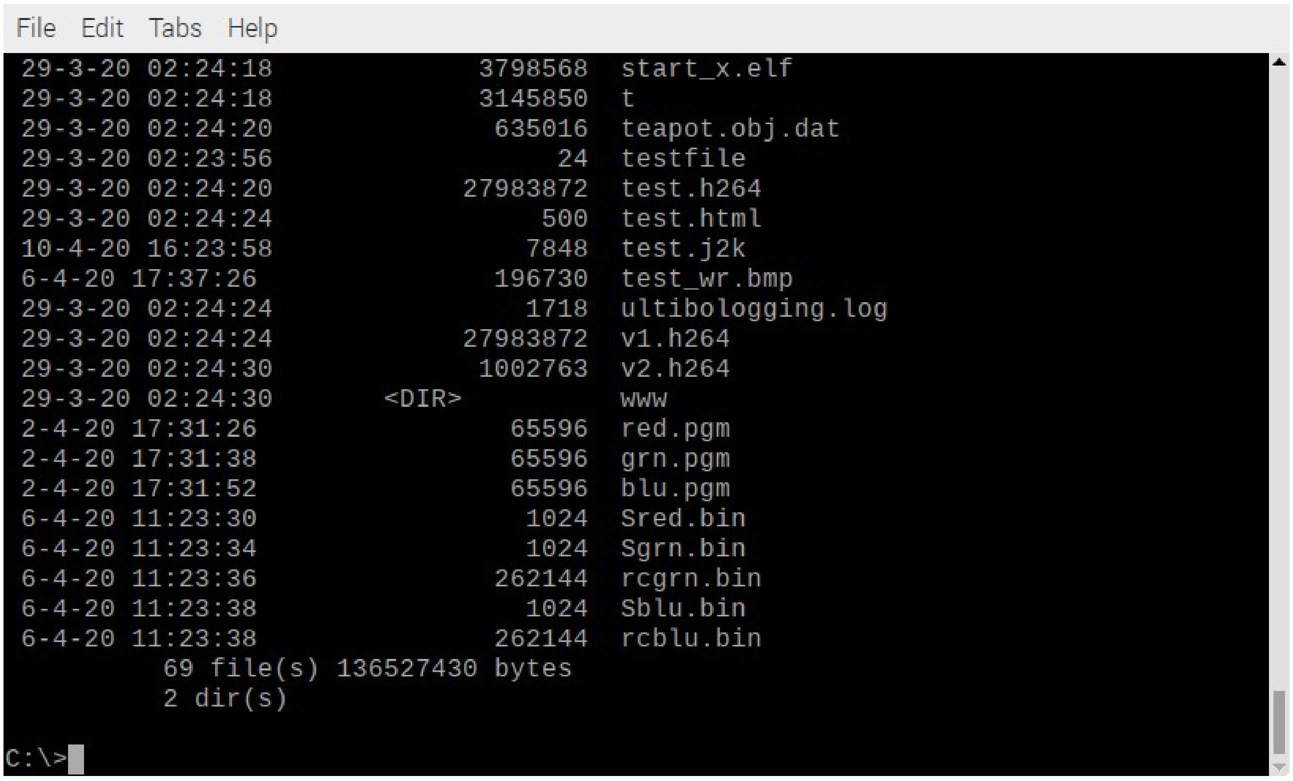
Decryption APICrypto.pas

shell1



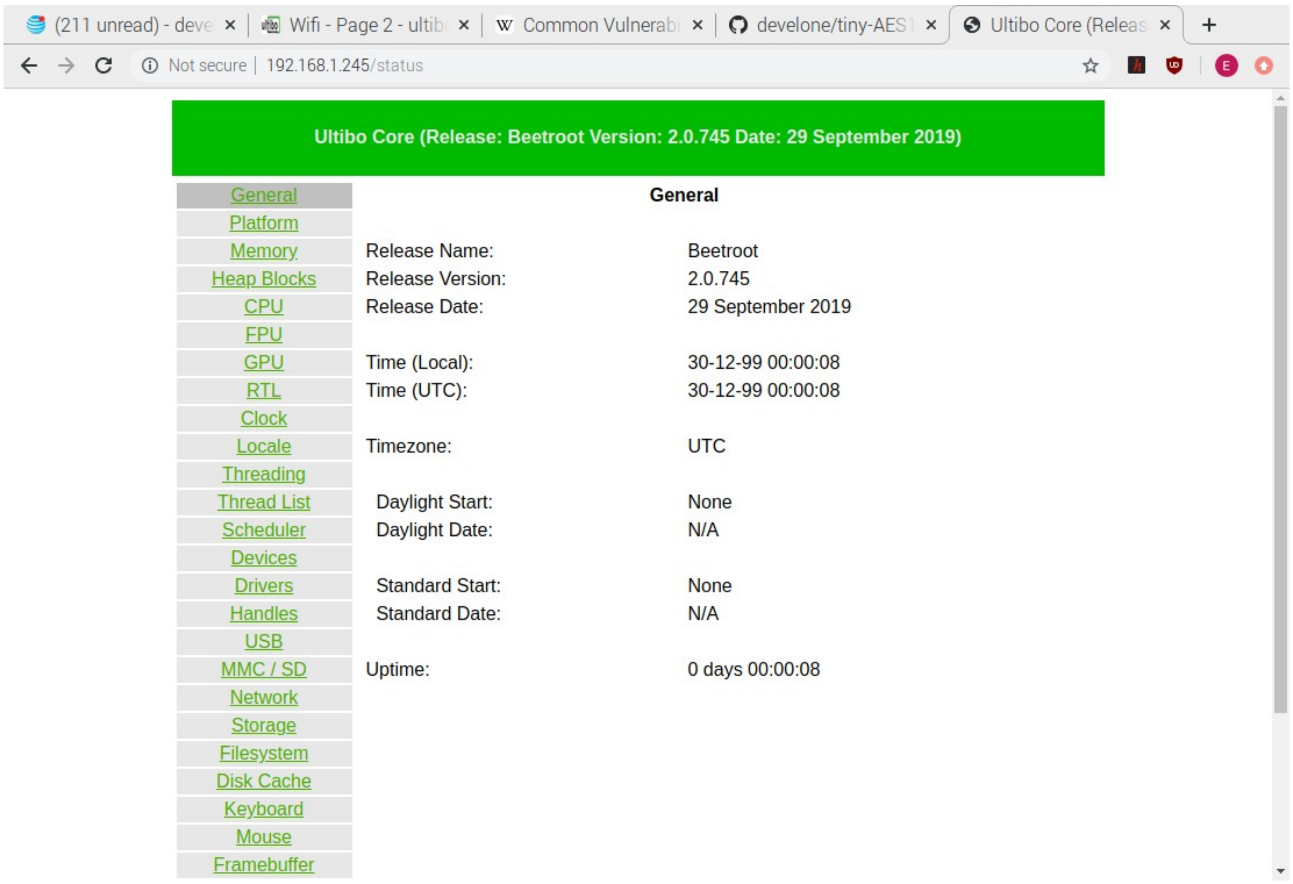
```
devel@mypi3-15:~/Ultibo_Projects/test_crypto/RPi2 $ telnet 192.168.1.245
```

shell2



```
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



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

General

Release Name:	Beetroot
Release Version:	2.0.745
Release Date:	29 September 2019
Time (Local):	30-12-99 00:00:08
Time (UTC):	30-12-99 00:00:08
Timezone:	UTC
Daylight Start:	None
Daylight Date:	N/A
Standard Start:	None
Standard Date:	N/A
Uptime:	0 days 00:00:08