

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

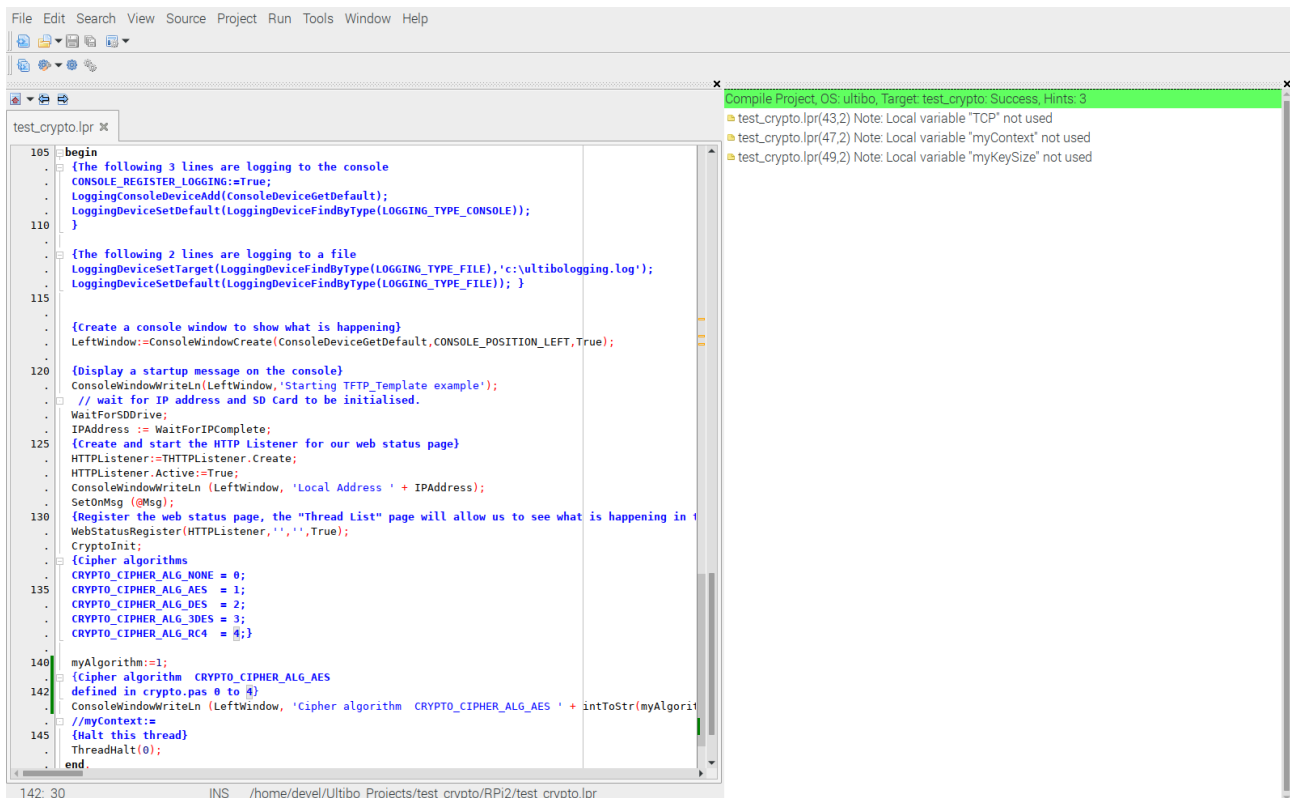
## crypto notes 05/04/20

### Starting with TFTP\_Template

\*\*\*\*\*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”.



```
File Edit Search View Source Project Run Tools Window Help
test_crypto.lpr x
105 begin
106 {The following 3 lines are logging to the console
107 . CONSOLE_REGISTER_LOGGING:=True;
108 . LoggingConsoleDeviceAdd(ConsoleDeviceGetDefault);
109 . LoggingDeviceSetDefault(LoggingDeviceFindByType(LOGGING_TYPE_CONSOLE));
110 }
111
112 {The following 2 lines are logging to a file
113 . LoggingDeviceSetTarget(LoggingDeviceFindByType(LOGGING_TYPE_FILE), 'c:\ultibo\logging.log');
114 . LoggingDeviceSetDefault(LoggingDeviceFindByType(LOGGING_TYPE_FILE)); }
115
116
117 {Create a console window to show what is happening}
118 . LeftWindow:=ConsoleWindowCreate(ConsoleDeviceGetDefault,CONSOLE_POSITION_LEFT,True);
119
120 {Display a startup message on the console}
121 . ConsoleWindowWriteLn(LeftWindow,'Starting TFTP_Template example');
122 . // wait for IP address and SD Card to be initialised.
123 . WaitForSDDrive;
124 . IPAddress := WaitForIPComplete;
125 {Create and start the HTTP Listener for our web status page}
126 . HTTPListener:=THTTPListener.Create;
127 . HTTPListener.Active:=True;
128 . ConsoleWindowWriteLn (LeftWindow, 'Local Address ' + IPAddress);
129 . SetOnMsg (@Msg);
130 {Register the web status page, the "Thread List" page will allow us to see what is happening in }
131 . WebStatusRegister(HTTPListener,','',True);
132 . CryptoInit;
133 {Cipher algorithms
134 . CRYPTO_CIPHER_ALG_NONE = 0;
135 . CRYPTO_CIPHER_ALG_AES = 1;
136 . CRYPTO_CIPHER_ALG_DES = 2;
137 . CRYPTO_CIPHER_ALG_3DES = 3;
138 . CRYPTO_CIPHER_ALG_RC4 = 4;}
139
140 myAlgorithm:=1;
141 {Cipher algorithm CRYPTO_CIPHER_ALG_AES
142 defined in crypto.pas 0 to 4}
143 . ConsoleWindowWriteLn (LeftWindow, 'Cipher algorithm CRYPTO_CIPHER_ALG_AES ' + IntToStr(myAlgorit
144 . //myContext:=
145 {Halt this thread}
146 . ThreadHalt(0);
147 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/dev/Ultibo\_Projects/test\_crypto/RPi2/test\_crypto.lpr

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

After adding APICrypto.pas

In test\_crypto.lpt in

**var**

**AESECBKey:PByte;**

**AESECBData:PByte;**

**AESECBAESKey:TAESKey;**

**AESCBCKey:PByte;**

**AESCBData:PByte;**

**AESCBCVector:PByte;**

**Cipher:PCipherContext;**

**key:String;**

**Data:String;**

**Actual:String;**

**PData:PString;**

**Datalen:LongWord;**

**InKey:LongWord;**

**InKeyStr:String;**

**InDataStr:String;**

**EncryptDecrypt:LongWord;**

**function**

**With the addition of function below matches APICrypto.pas**

**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)**

**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;**

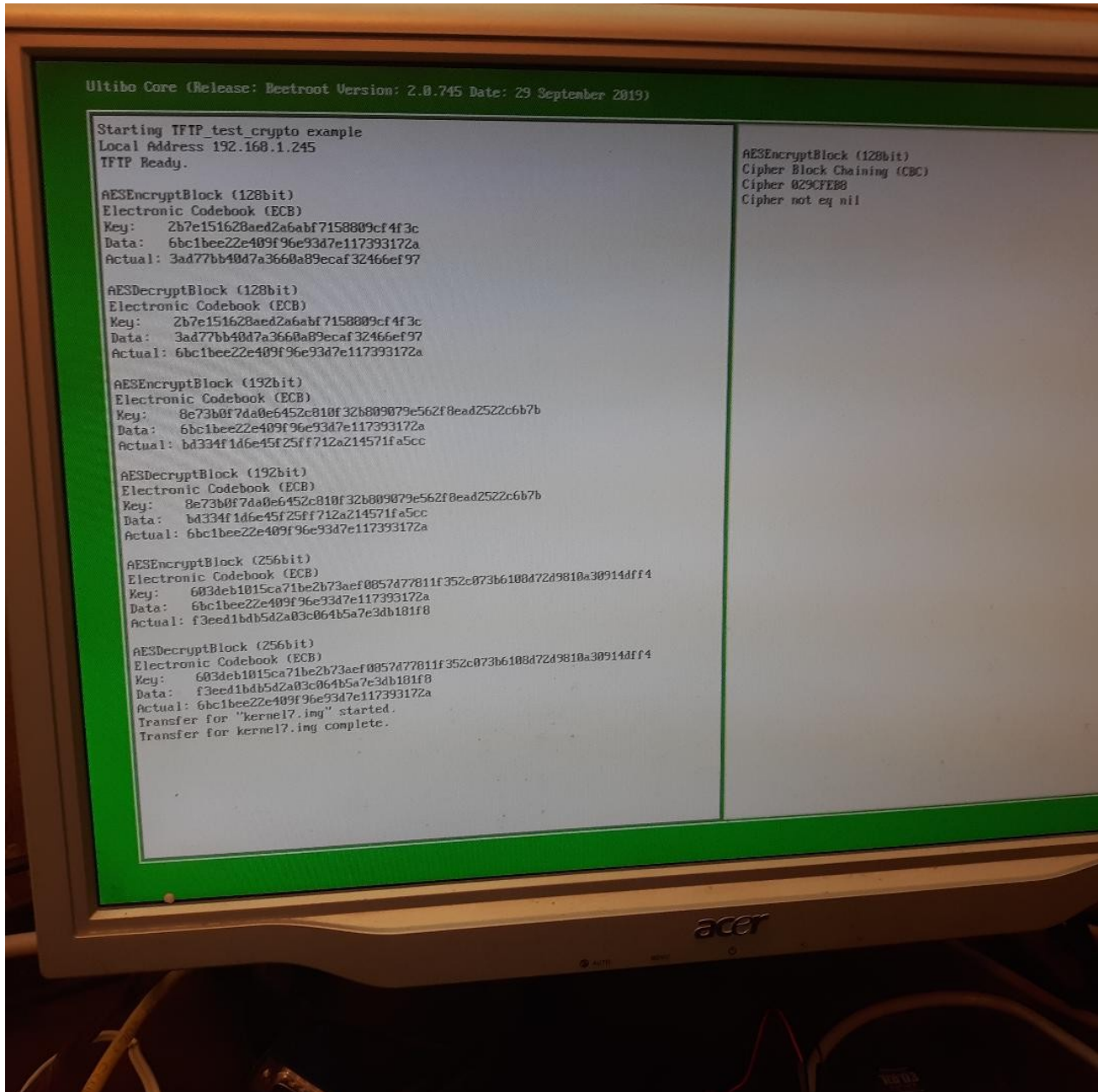
**if(EncryptDecrypt=0) then**

**begin**

**AESDecryptBlock(AESECBData,AESECBData,@AESECBAESKey);**

**end;**

./upker.sh



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

## Dencryption APICrypto.pas

(219 unread) - develone

Ultibo Core (Release: Bee

+

File | /home/dev/ultibo-web/Ultibo%20Core%20(R

Search

Star

Extensions

Menu

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

Actual: 6bc1bee22e409f96e93d7e117393172a

Result: Correct

encrdecr.jpeg

Show all

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

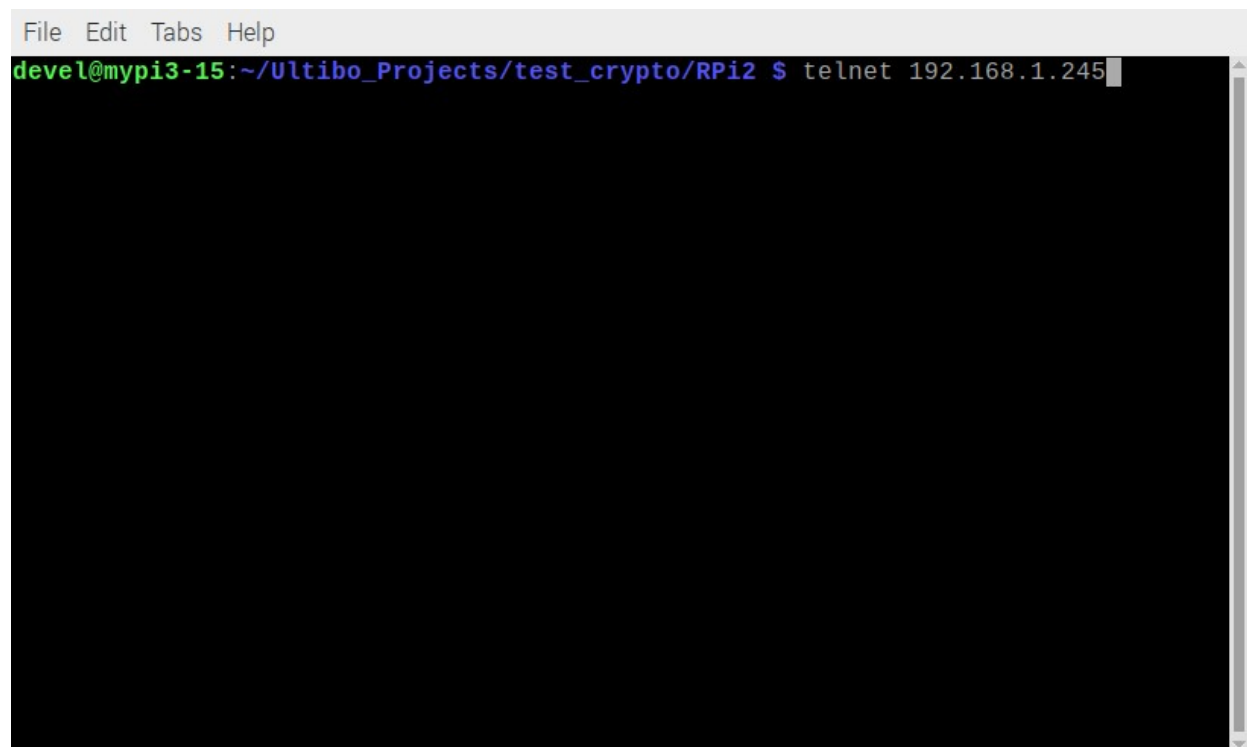
Actual: f3eed1bdb5d2a03c064b5a7e3db181f8

Result: Correct

AESECB.jpeg

Show all

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 displayed in green and blue text. The command 'telnet 192.168.1.245' is entered in white text, followed by a white cursor. A vertical scrollbar is visible on the right side of the terminal window.

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

shell2



