TNC Testing Form (REV1)	
Leaf on the Tree	Error Checking
Device Under Test (Testing Tree Number):	2.2.1
Date:	10/10/2020
Person(s) Conducting Experiment:	Kobe Keopraseuth
Signature:	
Experiment Purpose:	The purpose of this experiment is to verify an inputted packet has the correct FCS field, by performing a crc check on the other given subfields (Address, Control, PID, Info) as shown in the testing diagram.
Experiment Procedure:	To verify that it correctly verifies the FCS field, I made 2 testing array inputs, with a size greater than 120, which is the minimum. One array contains a correct FCS field and the other does not. Both arrays contain an input of 0x555555555555555555555555555555555555 (36 fives), excluding the FCS field. According the crc calculator, this input should have a crc output of 0x18c3. To verify that the generated crc is valid with the given input array, an online crc calculator was used.
Equipment Settings / Software Settings (w Revision):	Code will be implemented in Code Blocks IDE and it print out the input array, array after bit stuffing, the subfields obtained from the input array, FCS field in hexadecimal, crc calculation, and the result of whether the FCS field is valid or not.
Testing Diagram / Picture:	
Data Points:	**Online CRC Calculator:**

Testing Diagram tables:

Flag	Address	Control	Info	FCS	Flag
01111110	112/224 Bits	8/16 Bits	N*8 Bits	16 Bits	01111110

Figure 3.1a. U and S frame construction.

Flag	Address	Control	PID	Info	FCS	Flag
01111110	112/224 Bits	8/16 Bits	8 Bits	N*8 Bits	16 Bits	01111110

Data Points:

5555 5555 5555 5555 5555 5555 5555 5555 5555

Input type: ○ ASCII ◉ Hex   Output type: ◉ HEX

Algorithm	Result
CRC-16/X-25	0x18C3

Input1:

```
AX25 PACKET 1 = 0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1
,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,0,0,1,1,0,0,0,1,1,0,0,0,0,1,1,

AX.25 Packet(after bit suffing) = 0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0
,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,0,0,1,1,0,0,0,1,1,0,0,0,0,1,1,

address = 0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1
,1,0,1,0,1,0,1,0,1,0,1,

control = 0,1,0,1,0,1,0,1,

Contains a PID field!
PID = 0,1,0,1,0,1,0,1,

Info = 0,1,0,1,0,1,0,1,0,1,0,1,0,1,

FCS = 0,0,0,1,1,0,0,0,1,1,0,0,0,0,1,1,

FCS field in hex = 18c3
```

Output1:

```
crc = 18c3
out bit = 1
FCS is valid!!
PACKET IS VALID!!
```

Input2:

```
AX25 PACKET 2 = 0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,
,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,0,0,1,1,0,0,0,1,1,0,0,0,1,1,1,1,

AX.25 Packet(after bit suffing) = 0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0
,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,0,0,1,1,0,0,0,1,1,0,0,1,1,1,1,

address = 0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1
,1,0,1,0,1,0,1,0,1,0,1,

control = 0,1,0,1,0,1,0,1,

Contains a PID field!
PID = 0,1,0,1,0,1,0,1,

Info = 0,1,0,1,0,1,0,1,0,1,0,1,0,1,

FCS = 0,0,0,1,1,0,0,0,1,1,0,0,0,1,1,1,1,

FCS field in hex = 18cf
```

Ouput2:

```
crc = 18c3
out bit = 1
ERROR: FCS is invalid
ERROR: INVALID PACKET!!
```

Pass / Fail:	PASS
Interpreted Notes:	Code does verify the FCS field correctly. This was tested with different given FCS fields and different bits for the input array.
Recommendations for Modifications:	None