

# CAN Network

Project of Automotive Network Course

Students:

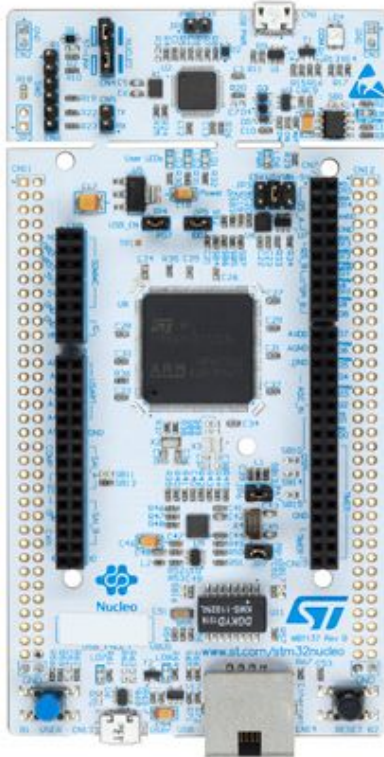
- Felipe Martins
- Lucas Cavalcanti
- Roberto Fernandes

# Summary

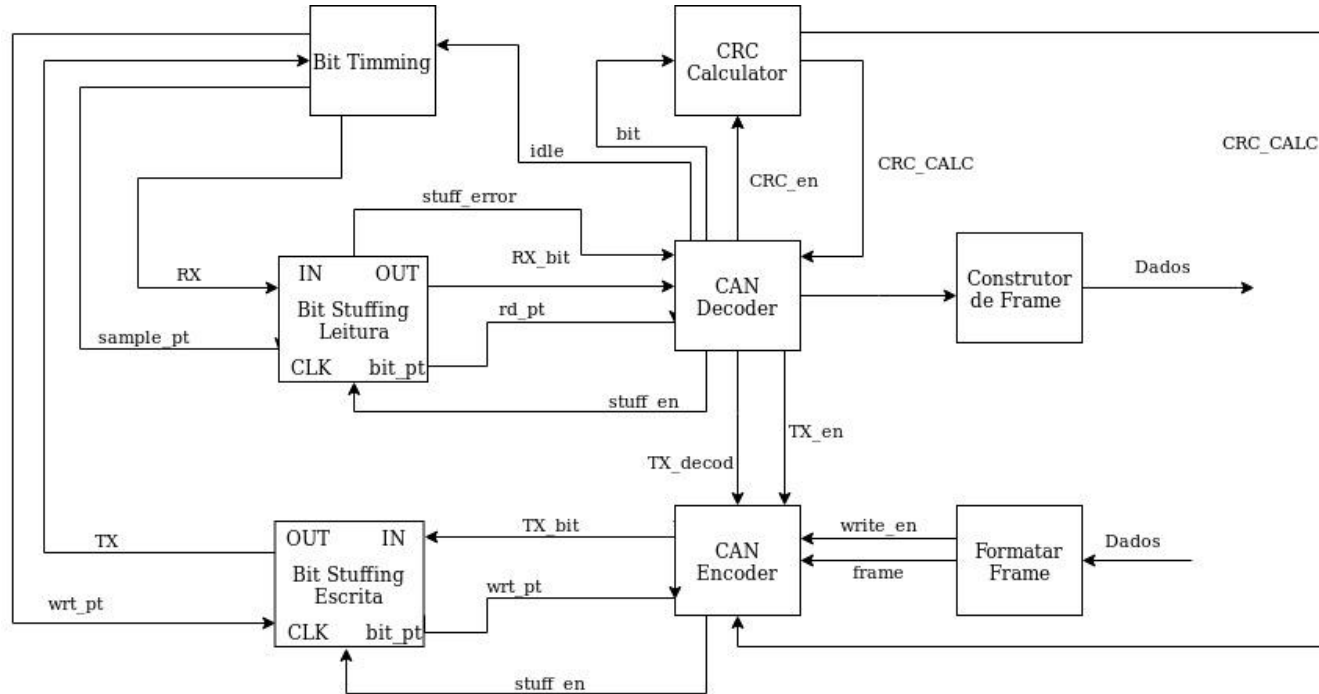
1. Platform
2. CAN Network
3. Bit Timing
4. Bit Stuffing
5. Decoder
6. CRC
7. Encoder
8. Frame Building
9. Encoder Tests
10. Decoder Tests



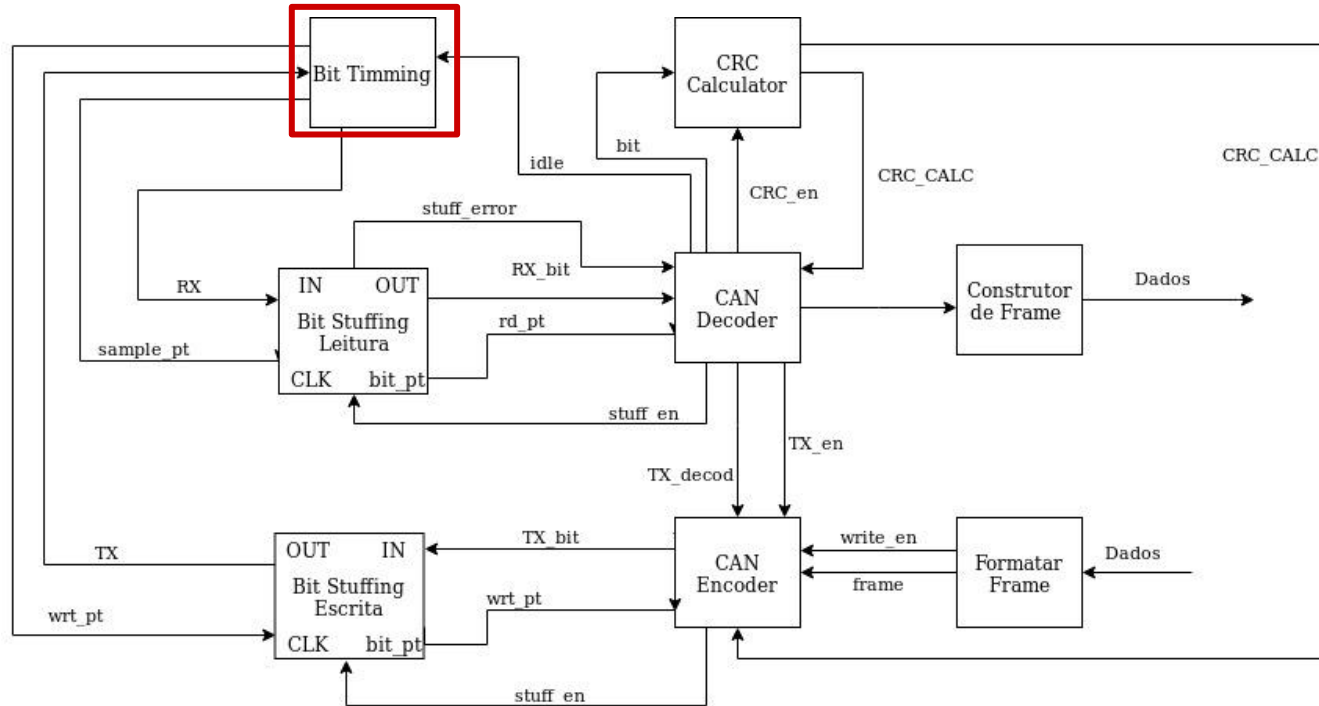
# STM32 - NUCLEO F767ZI



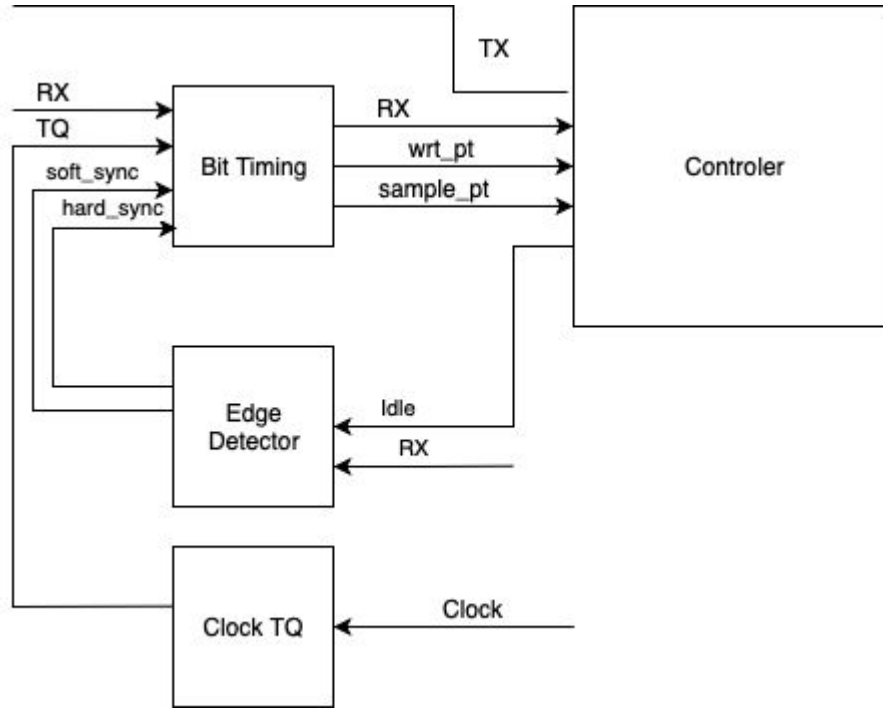
# Can Network



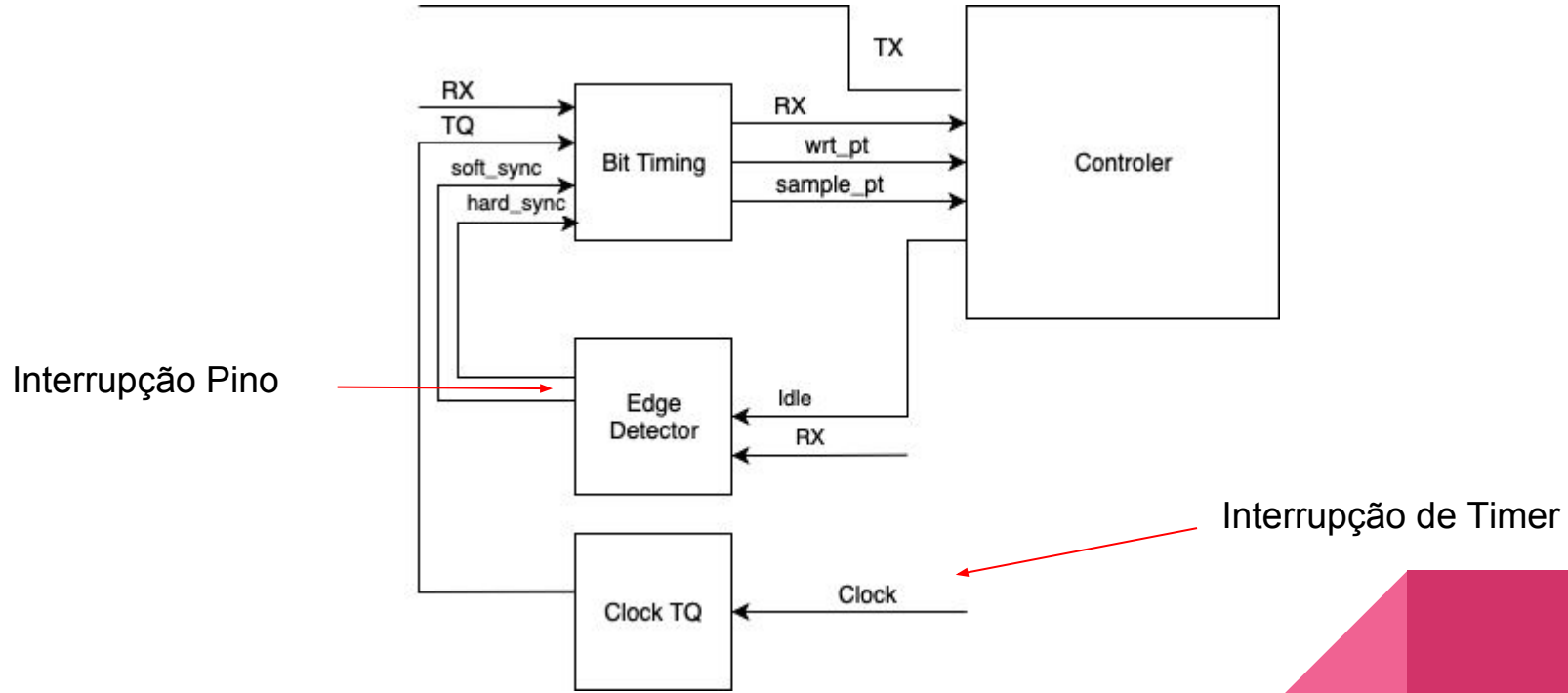
# Can Network



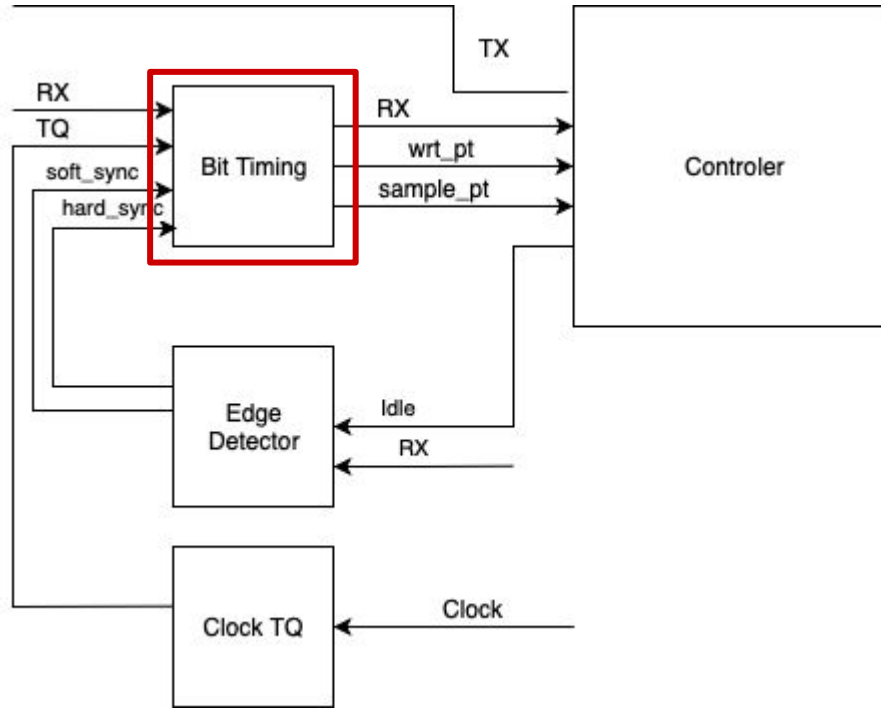
# Bit Timing



# Bit Timing

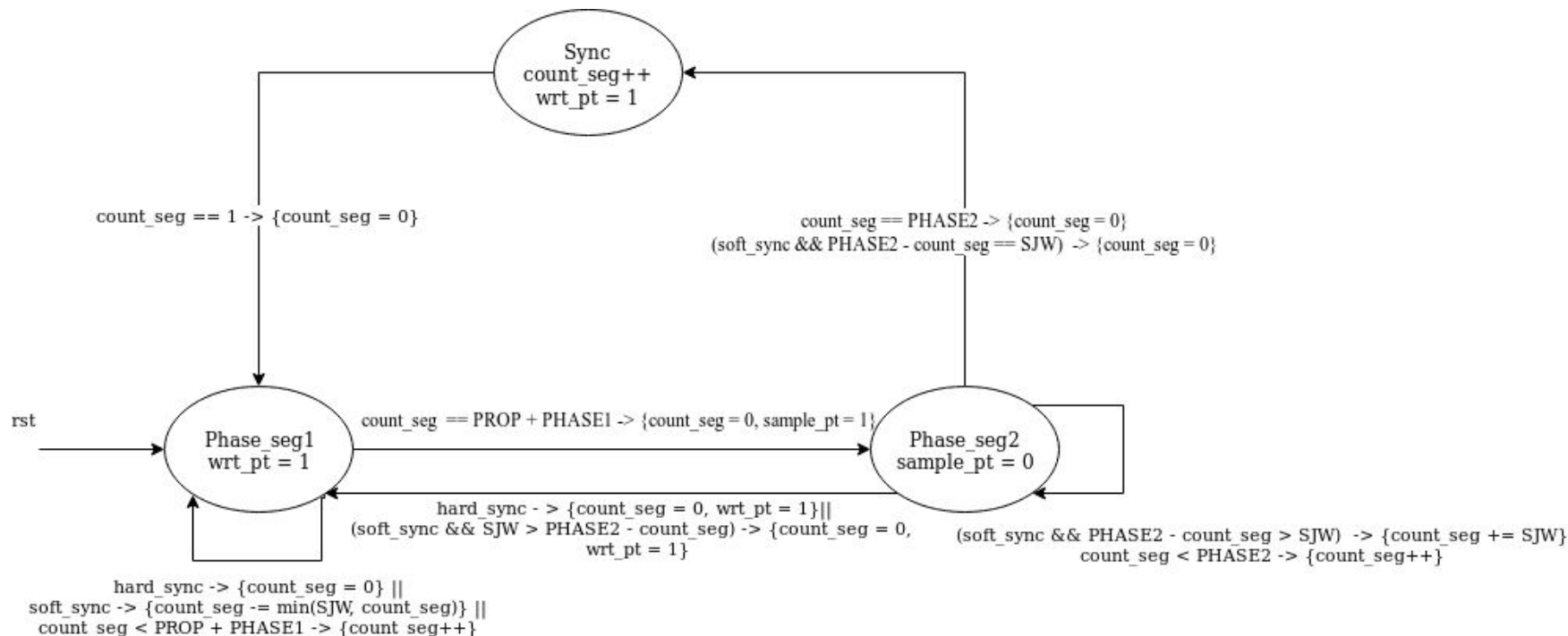


# Bit Timing





# Bit Timing



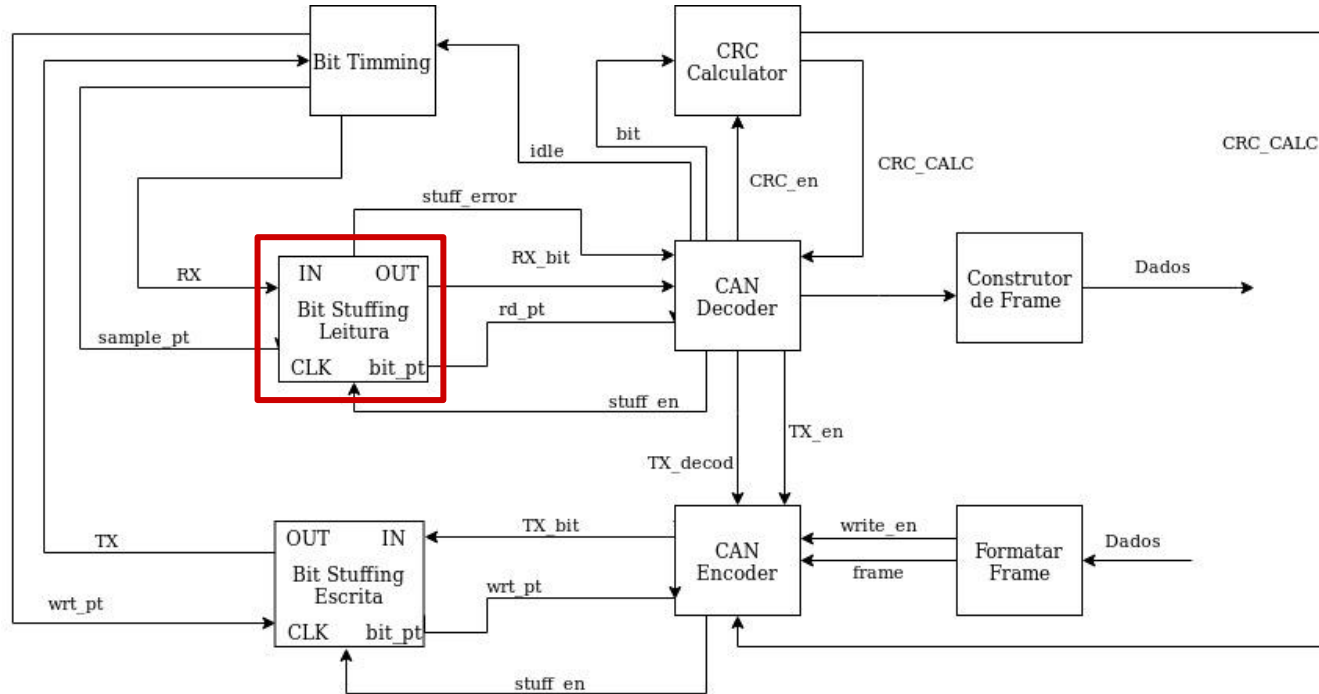
# Bit Timing

```
tq_clock.attach(bitTimingSM, TIME_QUANTA_S);
```

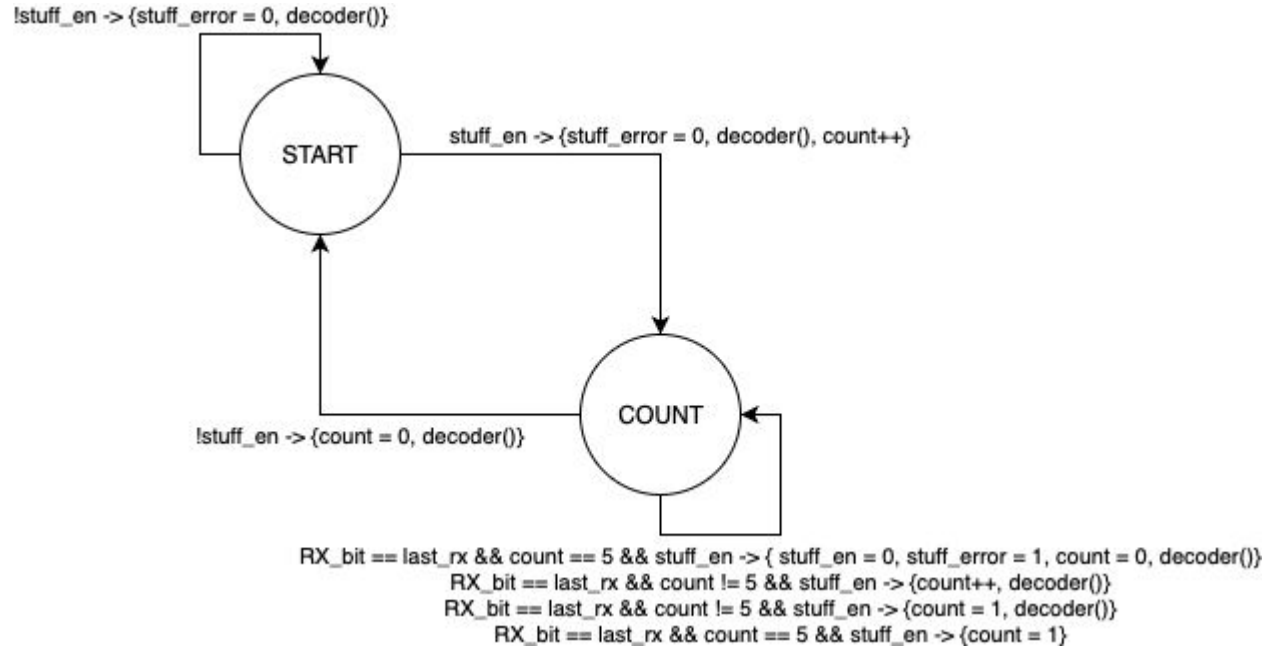
```
void bitTimingSM(){  
    static int state = PHASE1_ST;  
    static int count = 0;  
    switch(state){  
        case SYNC_ST: ...  
        case PHASE1_ST: ...  
        case PHASE2_ST: ...  
    }  
}
```



# Can Network



# Bit Stuffing

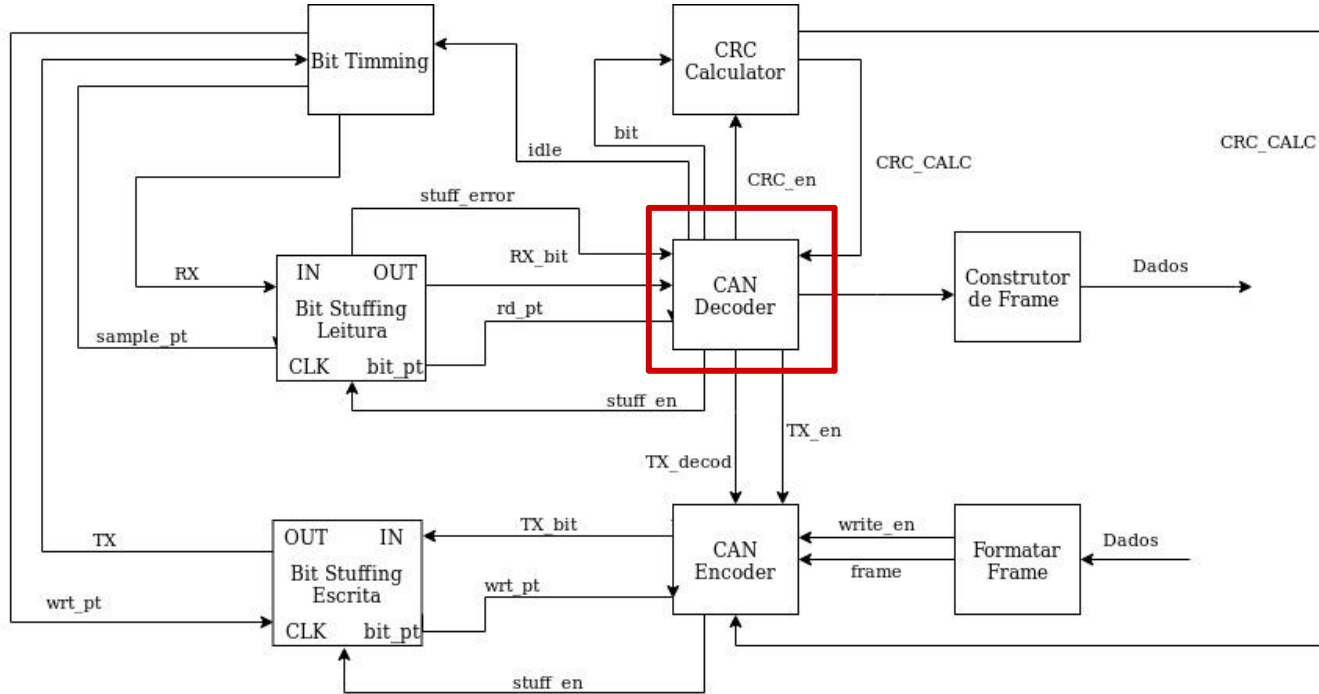


# Bit Stuffing

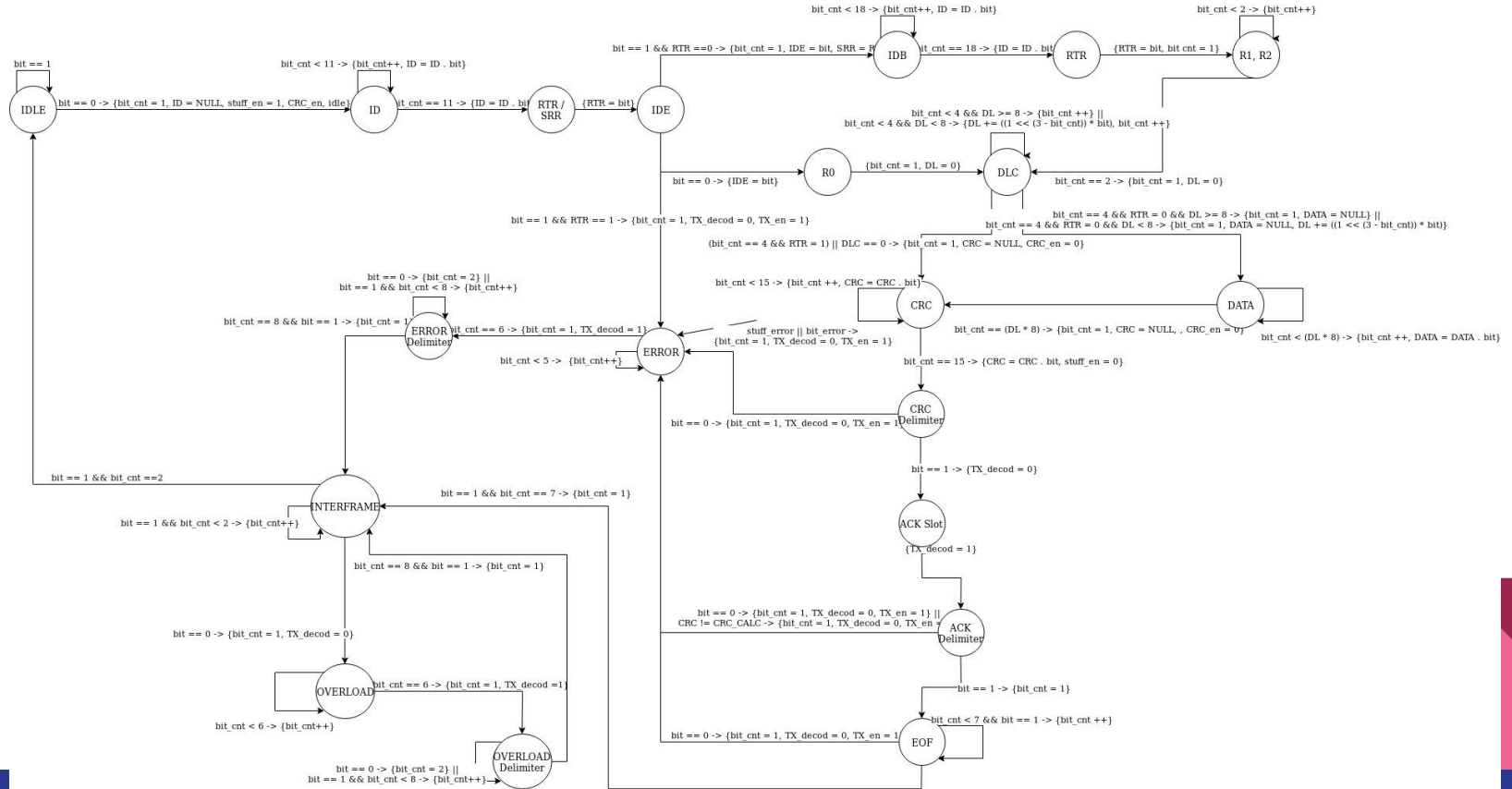
```
void bitstuffREAD()
{
    static int count = 0;
    static int state = 0;
    static int last_rx;
    last_rx = RX_bit;
    RX_bit = RX.read(); // TRANSCIEVER
    switch(state)
    {
        case(START): ...
        case(COUNT): ...
    }
}
```

```
case(START): ...
case(COUNT):
    if(RX_bit == last_rx && count == 5 && stuff_en)
    {
        stuff_en = 0;
        stuff_error = 1;
        count = 0;
        decoder();
        state = COUNT;
    } else {
        if(!stuff_en){ ...
        }
        else if(RX_bit == last_rx) { ...
        }
        else if(RX_bit != last_rx && count == 5) // STUFF
        {
            count = 1;
            debug(pc.printf("stuff - read\n"));
        }
        else if(RX_bit != last_rx && count != 5)
        {
            count = 1;
            decoder();
        }
    }
}
break;
```

# Can Network



# Decoder



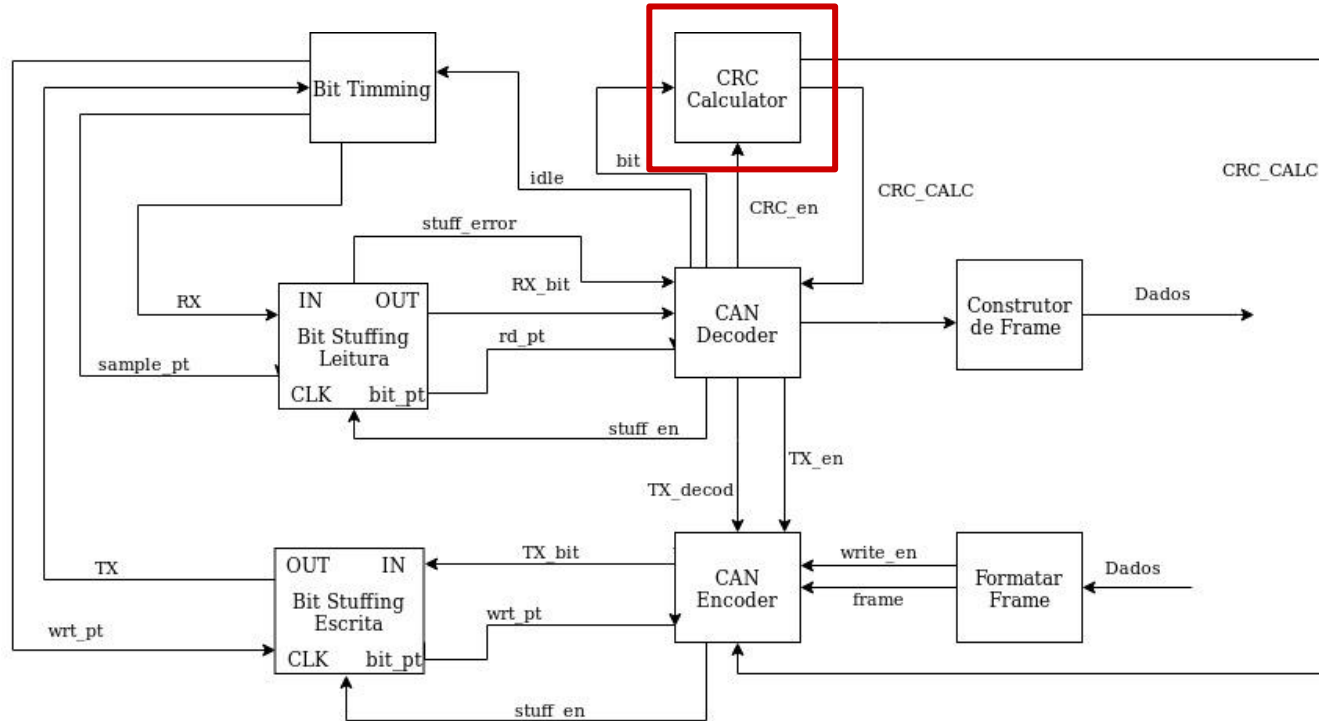
# Decoder

```
void decoder(){  
    static int state = 0;  
    static int bit_cnt = 0;  
    bool bit = RX_bit;  
    if(stuff_error || bit_error)  
    { ...  
    }
```

```
switch(state)  
{  
    case(IDLE): ...  
    case(ID): ...  
    case(SRR): ...  
    case(IDE): ...  
    case(R0): ...  
    case(IDB): ...  
    case(RTR): ...  
    case(R1): ...  
    case(R2): ...  
    case(DLC): ...  
    case(DATA): ...  
    case(CRC_V): ...  
    case(CRC_D): ...  
    case(ACK_S): ...  
    case(ACK_D): ...  
    case(EOFRAME): ...  
    case(INTERFRAME): ...  
    case(OVERLOAD): ...  
    case(OVERLOAD_D): ...  
    case(ERROR_FLAG): ...  
    case(ERROR_D): ...  
}  
calculateCRC(bit);  
}
```



# Can Network

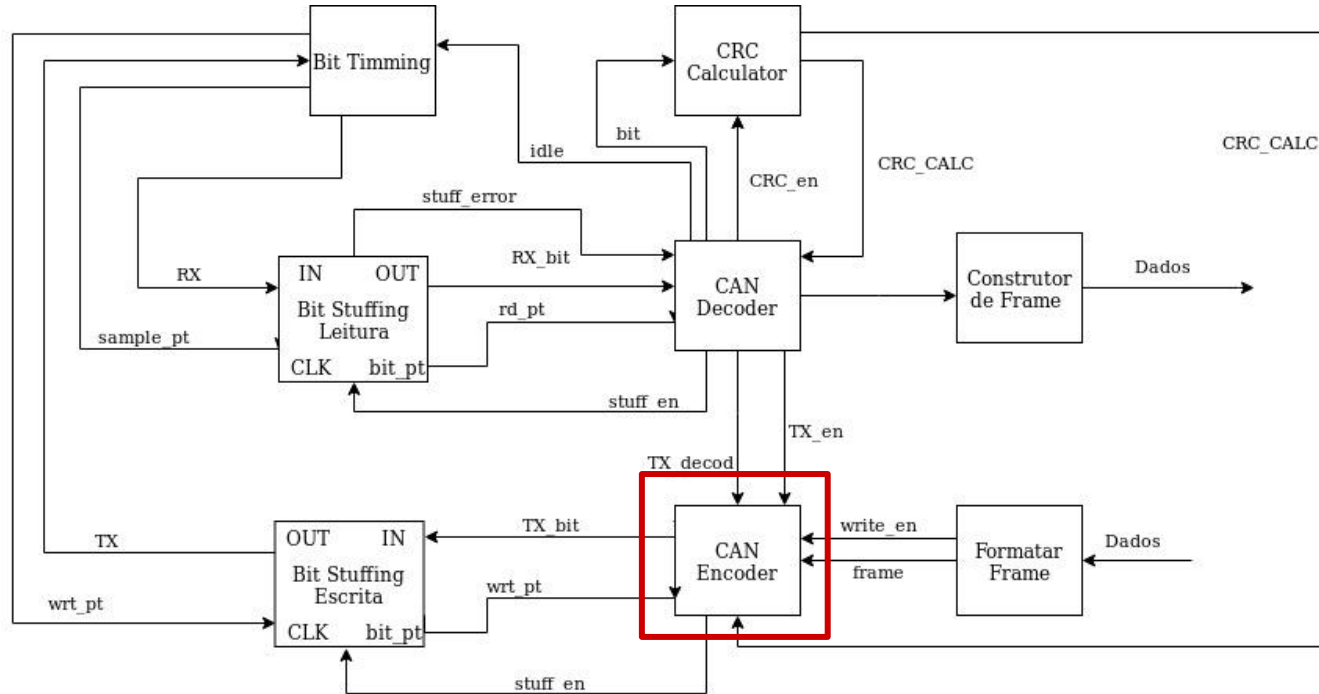


# CRC Calculator

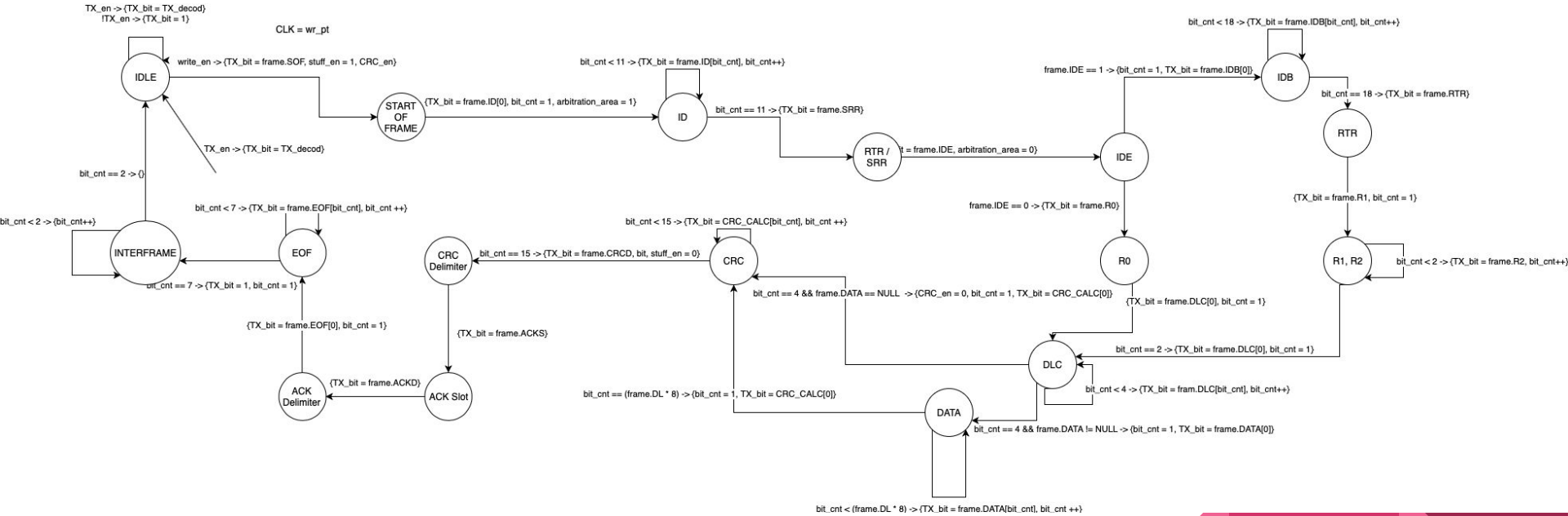
```
CRC_CALC = [15 bits = 0];
while(CRC_en){
    CRC_first = bit ^ CRC_CALC[14];
    CRC_CALC << 1;
    if(CRC_first){
        CRC_CALC = CRC_CALC ^ hex(4599);
    }
}
```

```
void calculateCRC(bool bit)
{
    if (CRC_en) {
        CRC_CALC <= 1;
        if ((CRC_CALC >= (1 << 15)) ^ bit) { // um smente no bit mais significativo
            CRC_CALC ^= 0x4599;
        }
        CRC_CALC &= 0x7fff; // zero no bit mais significativo e um no resto
    }
}
```

# Can Network



# Encoder



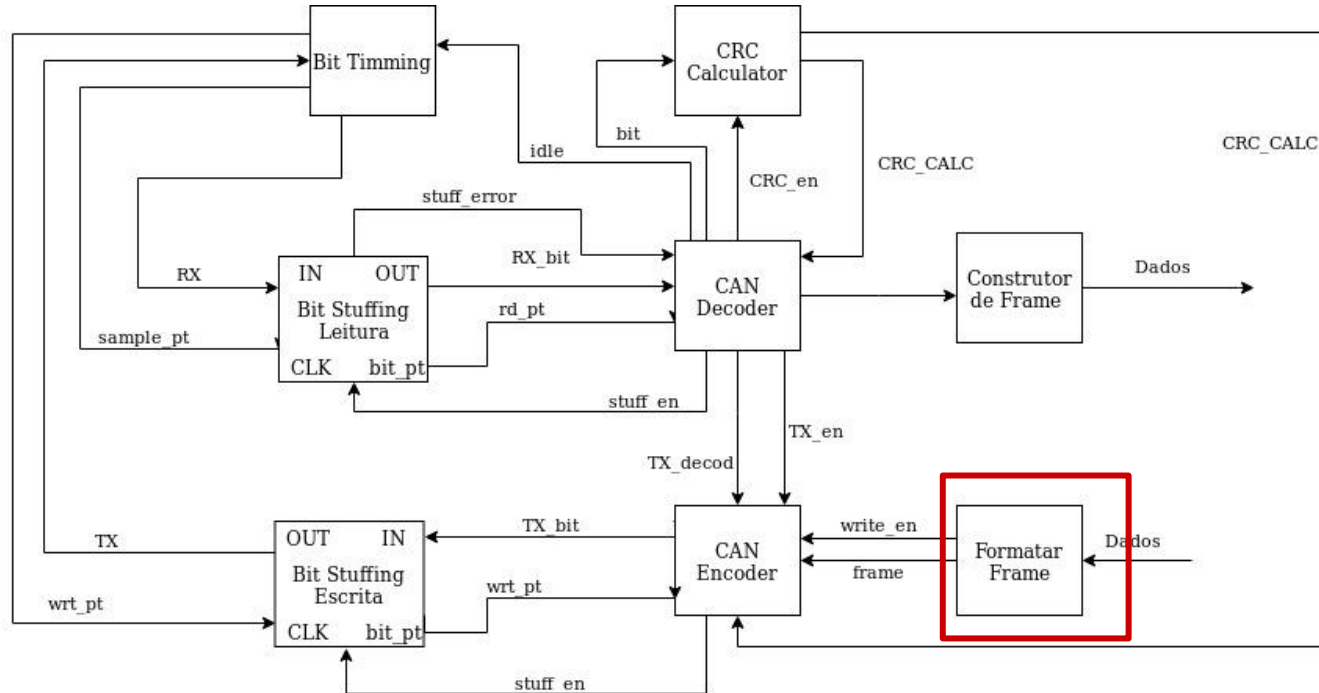
# Encoder

```
void encoder(){  
    static int state = 0;  
    static int bit_cnt = 0;  
    if(TX_en){  
        state = IDLE;  
    }  
}
```

```
switch(state){  
    case IDLE: ...  
    case SOF: ...  
    case ID: ...  
    case SRR: ...  
    case IDE: ...  
    case R0: ...  
    case IDB: ...  
    case RTR: ...  
    case R1: ...  
    case R2: ...  
    case DLC: ...  
    case DATA: ...  
    case CRC_V: ...  
    case CRC_D: ...  
    case ACK_S: ...  
    case ACK_D: ...  
    case EOFRAME: ...  
    case INTERFRAME: ...  
}
```

```
case R0:  
    if(RX_bit != TX_bit) //BIT_ERROR  
    {  
        state = IDLE;  
        TX_bit = 1;  
        bit_error = 1;  
        break;  
    }  
    TX_bit = (frame_send.DLC >> 3)&1;  
    bit_cnt = 1;  
    state = DLC;  
    break;
```

# Can Network



# Frame Building

- Struct used for frame parts.
- frame\_receive filled by Decoder.
- frame\_send read by Encoder.

- In Decoder

```
frame_rcv.ID = (frame_rcv.ID << 1) ^ bit;
```

- In Encoder

```
TX_bit = (frame_send.ID >> (10-bit_cnt)) & 1;
```

```
typedef struct CAN_FRAME
{
    bool SOF;
    uint16_t ID;
    bool SRR;
    bool RTR;
    bool IDE;
    bool R0;
    uint32_t IDB;
    bool R1;
    bool R2;
    uint8_t DLC;
    uint64_t DATA;
    bool data_b = false;
    uint16_t CRC_V;
    bool CRC_D;
    bool ACK_S;
    bool ACK_D;
    uint8_t EOFRAME;
} CAN_FRAME;
```

```
frame_send.SOF = 0;
frame_send.ID = 0x20;
frame_send.SRR = 0;
frame_send.RTR = 0;
frame_send.IDE = 0;
frame_send.R0 = 0;
frame_send.IDB = 0;
frame_send.R1 = 0;
frame_send.R2 = 0;
frame_send.DLC = 2;
frame_send.DATA = 0xaaaa;
frame_send.data_b = false;
frame_send.CRC_V = 30547;
frame_send.CRC_D = 1;
frame_send.ACK_S = 0;
frame_send.ACK_D = 1;
frame_send.EOFRAME = 127;
```

# Testing Encoder with Decoder

```
Printing Frame:
ID: 672
RTR: 0
IDE: 0
IDB: 0
SRR: 0
DLC: 8
DATA: aaaaaaaaaaaaaaaaaa
CRC_V: 81
CRC_D: 1
ACK_S: 0
ACK_D: 1
```

```
Sending.....
st machine: START OF FRAME
st machine: ID: 672
st machine: RTR/SRR: 0
st machine: IDE: 0
st machine: R0: 0
st machine: DLC: 8
st machine: DATA: aaaaaaaaaaaaaaaaaa
st machine: CRC Value: 81
st machine: CRC_CALC: 81
st machine: CRC_D: 1
st machine: ACK_S: 0
st machine: ACK_D: 1
st machine: EOFRAME
st machine: INTERFRAME
```

```
Printing Frames:
      Sent:      Receive:
ID: 672      672
RTR: 0      0
IDE: 0      0
IDB: 0      0
SRR: 0      0
DLC: 8      8
DATA: aaaaaaaaaaaaaaaaaa      aaaaaaaaaaaaaaaaaa
CRC_V: 81      81
CRC_D: 1      1
ACK_S: 0      0
ACK_D: 1      1
```

```
Printing Frame:
ID: 672
RTR: 0
IDE: 0
IDB: 0
SRR: 0
DLC: 7
DATA: aaaaaaaaaaaaaaaaaa
CRC_V: 22941
CRC_D: 1
ACK_S: 0
ACK_D: 1
```

```
Sending.....
st machine: START OF FRAME
st machine: ID: 672
st machine: RTR/SRR: 0
st machine: IDE: 0
st machine: R0: 0
st machine: DLC: 7
st machine: DATA: aaaaaaaaaaaaaaaaaa
st machine: CRC Value: 22941
st machine: CRC_CALC: 22941
st machine: CRC_D: 1
st machine: ACK_S: 0
st machine: ACK_D: 1
st machine: EOFRAME
st machine: INTERFRAME
```

```
Printing Frames:
      Sent:      Receive:
ID: 672      672
RTR: 0      0
IDE: 0      0
IDB: 0      0
SRR: 0      0
DLC: 7      7
DATA: aaaaaaaaaaaaaaaaaa      aaaaaaaaaaaaaaaaaa
CRC_V: 22941      22941
CRC_D: 1      1
ACK_S: 0      0
ACK_D: 1      1

st machine: IDLE
```

```
Printing Frame:
ID: 672
RTR: 0
IDE: 0
IDB: 0
SRR: 0
DLC: 3
DATA: aaaaaa
CRC_V: 9665
CRC_D: 1
ACK_S: 0
ACK_D: 1
```

```
Sending....
..
st machine: START OF FRAME
st machine: ID: 672
st machine: RTR/SRR: 0
st machine: IDE: 0
st machine: R0: 0
st machine: DLC: 3
st machine: DATA: aaaaaa
st machine: CRC Value: 9665
st machine: CRC_CALC: 9665
st machine: CRC_D: 1
st machine: ACK_S: 0
st machine: ACK_D: 1
st machine: EOFRAME
st machine: INTERFRAME
```

```
Printing Frames:
      Sent:      Receive:
ID: 672      672
RTR: 0      0
IDE: 0      0
IDB: 0      0
SRR: 0      0
DLC: 3      3
DATA: aaaaaa      aaaaaa
CRC_V: 9665      9665
CRC_D: 1      1
ACK_S: 0      0
ACK_D: 1      1

st machine: IDLE
```



# Testing Encoder with Decoder

Printing Frame:

ID: 672  
RTR: 0  
IDE: 0  
IDB: 0  
SRR: 0  
DLC: 0  
DATA: 0  
CRC\_V: 13013  
CRC\_D: 1  
ACK\_S: 0  
ACK\_D: 1

Sending....

..  
st machine: START OF FRAME

st machine: ID: 672  
st machine: RTR/SRR: 0  
st machine: IDE: 0  
st machine: R0: 0  
st machine: DLC: 0  
st machine: CRC Value: 13013  
st machine: CRC\_CALC: 13013  
st machine: CRC\_D: 1  
st machine: ACK\_S: 0  
st machine: ACK\_D: 1  
st machine: EOFRAME  
st machine: INTERFRAME

Printing Frames:

Sent:	Receive:
ID: 672	672
RTR: 0	0
IDE: 0	0
IDB: 0	0
SRR: 0	0
DLC: 0	0
DATA: 0	0
CRC_V: 13013	13013
CRC_D: 1	1
ACK_S: 0	0
ACK_D: 1	1

st machine: IDLE

Printing Frame:

ID: 672  
RTR: 1  
IDE: 0  
IDB: 0  
SRR: 0  
DLC: 0  
DATA: 0  
CRC\_V: 16656  
CRC\_D: 1  
ACK\_S: 0  
ACK\_D: 1

Sending....

..  
st machine: START OF FRAME

st machine: ID: 672  
st machine: RTR/SRR: 1  
st machine: IDE: 0  
st machine: R0: 0  
st machine: DLC: 0  
st machine: CRC Value: 16656  
st machine: CRC\_CALC: 16656  
st machine: CRC\_D: 1  
st machine: ACK\_S: 0  
st machine: ACK\_D: 1  
st machine: EOFRAME  
st machine: INTERFRAME

Printing Frames:

Sent:	Receive:
ID: 672	672
RTR: 1	1
IDE: 0	0
IDB: 0	0
SRR: 0	0
DLC: 0	0
DATA: 0	0
CRC_V: 16656	16656
CRC_D: 1	1
ACK_S: 0	0
ACK_D: 1	1

st machine: IDLE

Printing Frame:

ID: 672  
RTR: 1  
IDE: 0  
IDB: 0  
SRR: 0  
DLC: 1  
DATA: 0  
CRC\_V: 1161  
CRC\_D: 1  
ACK\_S: 0  
ACK\_D: 1

Sending....

..  
st machine: START OF FRAME

st machine: ID: 672  
st machine: RTR/SRR: 1  
st machine: IDE: 0  
st machine: R0: 0  
st machine: DLC: 1  
st machine: CRC Value: 1161  
st machine: CRC\_CALC: 1161  
st machine: CRC\_D: 1  
st machine: ACK\_S: 0  
st machine: ACK\_D: 1  
st machine: EOFRAME  
st machine: INTERFRAME

Printing Frames:

Sent:	Receive:
ID: 672	672
RTR: 1	1
IDE: 0	0
IDB: 0	0
SRR: 0	0
DLC: 1	1
DATA: 0	0
CRC_V: 1161	1161
CRC_D: 1	1
ACK_S: 0	0
ACK_D: 1	1

st machine: IDLE

# Testing Encoder with Decoder

```
Printing Frame:
ID: 449
RTR: 0
IDE: 1
IDB: 3007a
SRR: 1
DLC: 8
DATA: aaaaaaaaaaaaaa
CRC_V: 31733
CRC_D: 1
ACK_S: 0
ACK_D: 1
```

Sending....

```
..
st machine: START OF FRAME
st machine: ID: 449
st machine: RTR/SRR: 1
st machine: IDE: 1
st machine: SRR: 1
st machine: IDB: 3007a
st machine: RTR: 0
st machine: R1: 0
st machine: R2: 0
st machine: DLC: 8
st machine: DATA: aaaaaaaaaaaaaa
st machine: CRC Value: 31733
st machine: CRC CALC: 31733
st machine: CRC D: 1
st machine: ACK S: 0
st machine: ACK D: 1
st machine: EOFFRAME
st machine: INTERFRAME
```

Printing Frames:

Sent:	Receive:
ID: 449	449
RTR: 0	0
IDE: 1	1
IDB: 3007a	3007a
SRR: 1	1
DLC: 8	8
DATA: aaaaaaaaaaaaaa	aaaaaaaaaaaaa
CRC_V: 31733	31733
CRC_D: 1	1
ACK_S: 0	0
ACK_D: 1	1

st machine: IDLE

```
Printing Frame:
ID: 449
RTR: 1
IDE: 1
IDB: 3007a
SRR: 1
DLC: 8
DATA: 0
CRC_V: 10742
CRC_D: 1
ACK_S: 0
ACK_D: 1
```

Sending....

```
..
st machine: START OF FRAME
st machine: ID: 449
st machine: RTR/SRR: 1
st machine: IDE: 1
st machine: SRR: 1
st machine: IDB: 3007a
st machine: RTR: 1
st machine: R1: 0
st machine: R2: 0
st machine: DLC: 8
st machine: CRC Value: 10742
st machine: CRC CALC: 10742
st machine: CRC D: 1
st machine: ACK S: 0
st machine: ACK D: 1
st machine: EOFFRAME
st machine: INTERFRAME
```

Printing Frames:

Sent:	Receive:
ID: 449	449
RTR: 1	1
IDE: 1	1
IDB: 3007a	3007a
SRR: 1	1
DLC: 8	8
DATA: 0	0
CRC_V: 10742	10742
CRC_D: 1	1
ACK_S: 0	0
ACK_D: 1	1

st machine: IDLE

```
Printing Frame:
ID: 3
RTR: 0
IDE: 1
IDB: 0
SRR: 1
DLC: 15
DATA: ffffffffffffffff
CRC_V: 20214
CRC_D: 1
ACK_S: 0
ACK_D: 1
```

Sending....

```
..
st machine: START OF FRAME
st machine: ID: 3
st machine: RTR/SRR: 1
st machine: IDE: 1
st machine: SRR: 1
st machine: IDB: 0
st machine: RTR: 0
st machine: R1: 0
st machine: R2: 0
st machine: DLC: 15 --> DLC: 8
st machine: DATA: ffffffffffffffff
st machine: CRC Value: 20214
st machine: CRC CALC: 20214
st machine: CRC D: 1
st machine: ACK S: 0
st machine: ACK D: 1
st machine: EOFFRAME
st machine: INTERFRAME
```

Printing Frames:

Sent:	Receive:
ID: 3	3
RTR: 0	0
IDE: 1	1
IDB: 0	0
SRR: 1	1
DLC: 15	8
DATA: ffffffffffffffff	fffffffffffffff
CRC_V: 20214	20214
CRC_D: 1	1
ACK_S: 0	0
ACK_D: 1	1

st machine: IDLE

# Testing Encoder with Decoder Repeating

```
Sending....  
  
Printing Frames:  
      Sent:      Receive:  
ID:   672        672  
RTR:   0         0  
IDE:   0         0  
IDB:   0         0  
SRR:   0         0  
DLC:   3         3  
DATA: aaaaaa   aaaaaa  
CRC_V: 9665     9665  
CRC_D: 1        1  
ACK_S: 0        0  
ACK_D: 1        1  
  
Sending....  
  
Printing Frames:  
      Sent:      Receive:  
ID:   672        672  
RTR:   0         0  
IDE:   0         0  
IDB:   0         0  
SRR:   0         0  
DLC:   3         3  
DATA: aaaaaa   aaaaaa  
CRC_V: 9665     9665  
CRC_D: 1        1  
ACK_S: 0        0  
ACK_D: 1        1
```

```
Sending....  
  
Printing Frames:  
      Sent:      Receive:  
ID:   672        672  
RTR:   0         0  
IDE:   0         0  
IDB:   0         0  
SRR:   0         0  
DLC:   3         3  
DATA: aaaaaa   aaaaaa  
CRC_V: 9665     9665  
CRC_D: 1        1  
ACK_S: 0        0  
ACK_D: 1        1  
  
Sending....  
  
Printing Frames:  
      Sent:      Receive:  
ID:   672        672  
RTR:   0         0  
IDE:   0         0  
IDB:   0         0  
SRR:   0         0  
DLC:   3         3  
DATA: aaaaaa   aaaaaa  
CRC_V: 9665     9665  
CRC_D: 1        1  
ACK_S: 0        0  
ACK_D: 1        1
```

# Testing Decoder Repeating Frame

```
.....
st machine: START OF FRAME
st machine: ID: 3
st machine: RTR/SRR: 1
st machine: IDE: 1
st machine: SRR: 1
st machine: IDB: 0
st machine: RTR: 0
st machine: R1: 0
st machine: R2: 0
st machine: DLC: 15 --> DLC: 8
st machine: DATA: ffffffffffffffff
st machine: CRC Value: 11311
st machine: CRC_CALC: 11311
st machine: CRC_D: 1
st machine: ACK_S: 0
st machine: ACK_D: 1
st machine: EOFRAME
st machine: INTERFRAME

Printing Frame:
ID: 3
RTR: 0
IDE: 1
IDB: 0
SRR: 1
DLC: 8
DATA: ffffffffffffffff
CRC_V: 11311
CRC_D: 1
ACK_S: 0
ACK_D: 1

st machine: IDLE
.....
st machine: START OF FRAME
st machine: ID: 3
st machine: RTR/SRR: 1
st machine: IDE: 1
st machine: SRR: 1
st machine: IDB: 0
st machine: RTR: 0
st machine: R1: 0
st machine: R2: 0
st machine: DLC: 15 --> DLC: 8
st machine: DATA: ffffffffffffffff
st machine: CRC Value: 11311
st machine: CRC_CALC: 11311
st machine: CRC_D: 1
st machine: ACK_S: 0
st machine: ACK_D: 1
st machine: EOFRAME
st machine: INTERFRAME
```

# Testing Decoder with Error

```
st machine: START OF FRAME
st machine: ID: 672
st machine: RTR/SRR: 0
st machine: IDE: 0
st machine: R0: 0
st machine: DLC: 8
st machine: DATA: aaaaaaaaaaaaaaaaaa
st machine: CRC_Value: 81
st machine: CRC_CALC: 81
st machine: CRC_D: 1
st machine: ACK_S: 1
st machine: ACK_D: 0
st machine: ACK Error
st machine: ERROR_FLAG
st machine: ERROR_DELIMITER
st machine: INTERFRAME
```

Printing Frame:

```
ID: 672
RTR: 0
IDE: 0
IDB: 0
SRR: 0
DLC: 8
DATA: aaaaaaaaaaaaaaaaaa
CRC_V: 81
CRC_D: 1
ACK_S: 1
ACK_D: 0
```

st machine: IDLE

```
st machine: START OF FRAME
st machine: ID: 672
st machine: RTR/SRR: 0
st machine: IDE: 0
st machine: R0: 0
st machine: DLC: 8
st machine: DATA: aaaaaaaaaaaaaaaaaa
st machine: Error Detected: STUFF_ERROR
st machine: ERROR_FLAG
st machine: ERROR_DELIMITER
st machine: INTERFRAME
```

Printing Frame:

```
ID: 672
RTR: 0
IDE: 0
IDB: 0
SRR: 0
DLC: 8
DATA: aaaaaaaaaaaaaaaaaa
CRC_V: 0
CRC_D: 0
ACK_S: 0
ACK_D: 0
```

st machine: IDLE

```
st machine: START OF FRAME
st machine: ID: 672
st machine: RTR/SRR: 0
st machine: IDE: 0
st machine: R0: 0
st machine: DLC: 8
st machine: DATA: aaaaaaaaaaaaaaaaaa
st machine: CRC_Value: 113
st machine: CRC_CALC: 81
st machine: CRC_D: 1
st machine: ACK_S: 0
st machine: ACK_D: 1
st machine: CRC_V != CRC_CALC
st machine: ERROR_FLAG
st machine: ERROR_DELIMITER
st machine: INTERFRAME
```

Printing Frame:

```
ID: 672
RTR: 0
IDE: 0
IDB: 0
SRR: 0
DLC: 8
DATA: aaaaaaaaaaaaaaaaaa
CRC_V: 113
CRC_D: 1
ACK_S: 0
ACK_D: 1
```

st machine: IDLE

# Working Schedule

	Felipe Martins	Lucas Cavalcanti	Roberto Fernandes
Bit Timing	  		
Bit Stuffing			  
Encoder	 	 	 
Decoder	 	 	 
Integration	  	 	  
Validation		  	
Reports	  	 	  

# Problems Found

- Transceiver not working.
- Implementation in a single core microcontroller.
- No ground truth implemented.
- Debugging interface is slow.
- Microcontroller does not support 500Kbps CAN Network, even with 216MHz.
- Implementation of state machines need to respect the priorities.
- One state machine interfere in others.



# Conclusion

- CAN Network should be implemented in FPGA.
- Implementing CAN Network in C/C++ improves its comprehension.
- CAN Network has an intelligent strategy for synchronizing.

