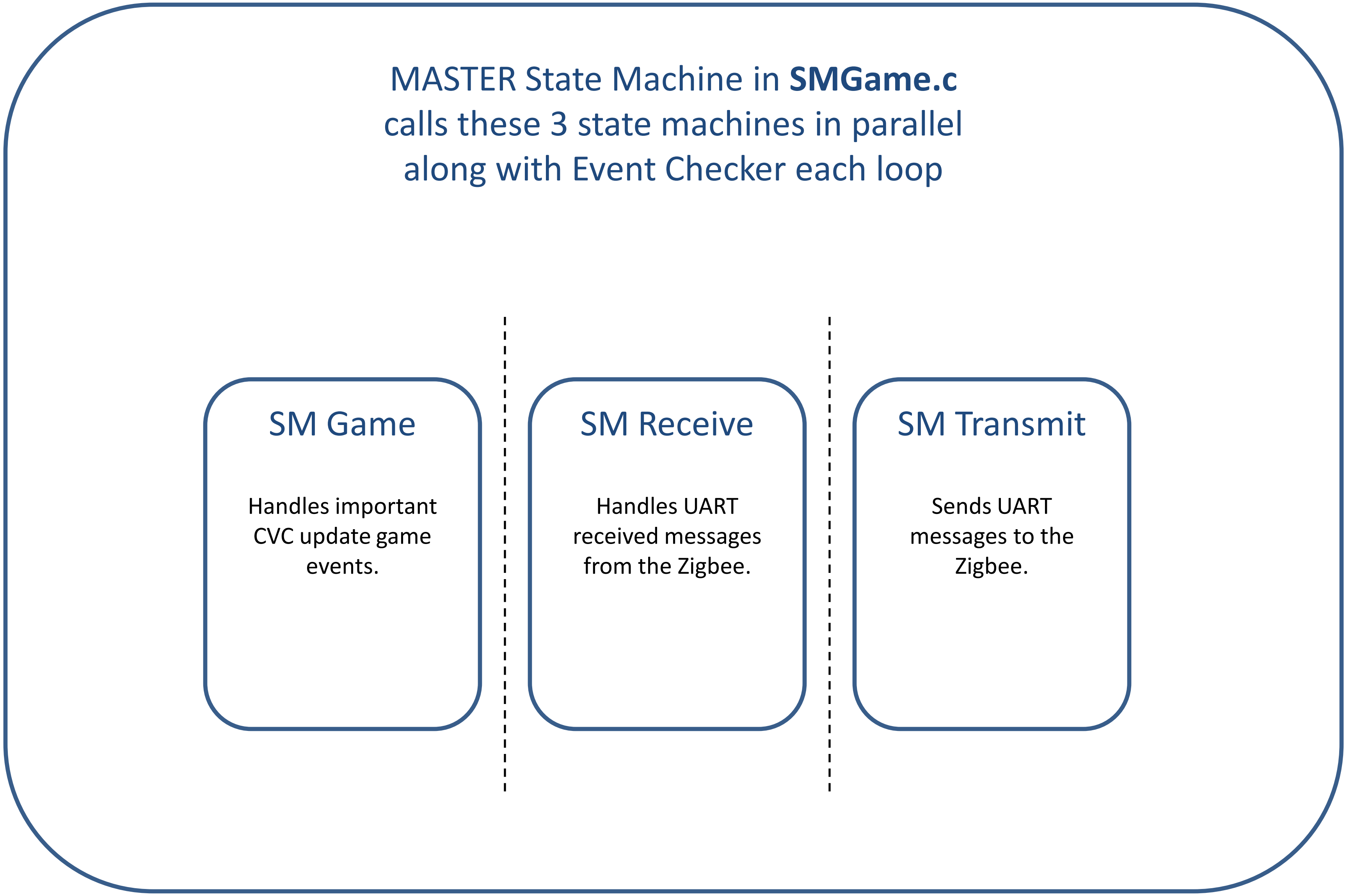
CVC E128: STATE MACHINES



MASTER State Machine in **SMGame.c**
calls these 3 state machines in parallel
along with Event Checker each loop

SM Game

Handles important CVC update game events.

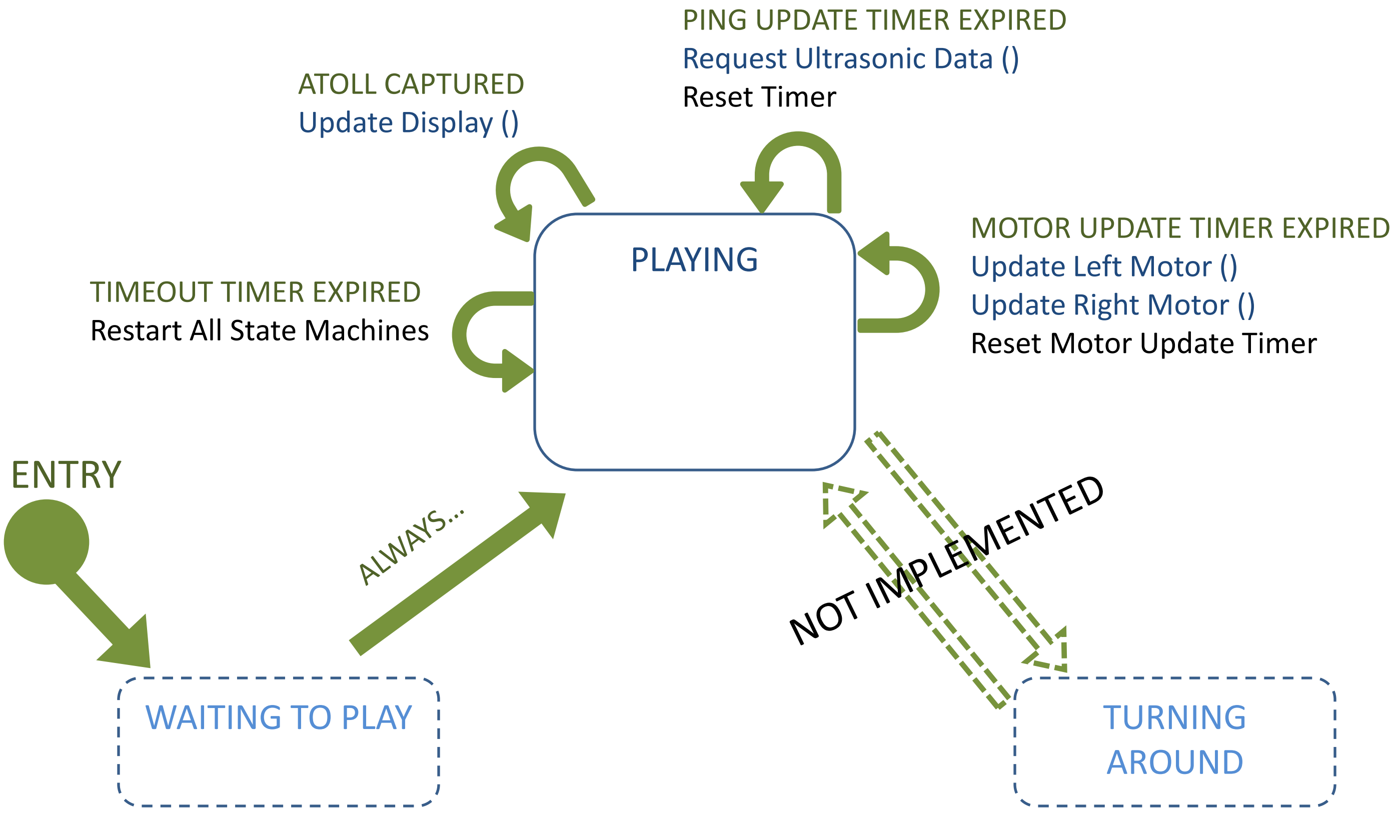
SM Receive

Handles UART received messages from the Zigbee.

SM Transmit

Sends UART messages to the Zigbee.

SM GAME



ATOLL CAPTURED
Update Display ()

PING UPDATE TIMER EXPIRED
Request Ultrasonic Data ()
Reset Timer

TIMEOUT TIMER EXPIRED
Restart All State Machines

MOTOR UPDATE TIMER EXPIRED
Update Left Motor ()
Update Right Motor ()
Reset Motor Update Timer

PLAYING

ENTRY

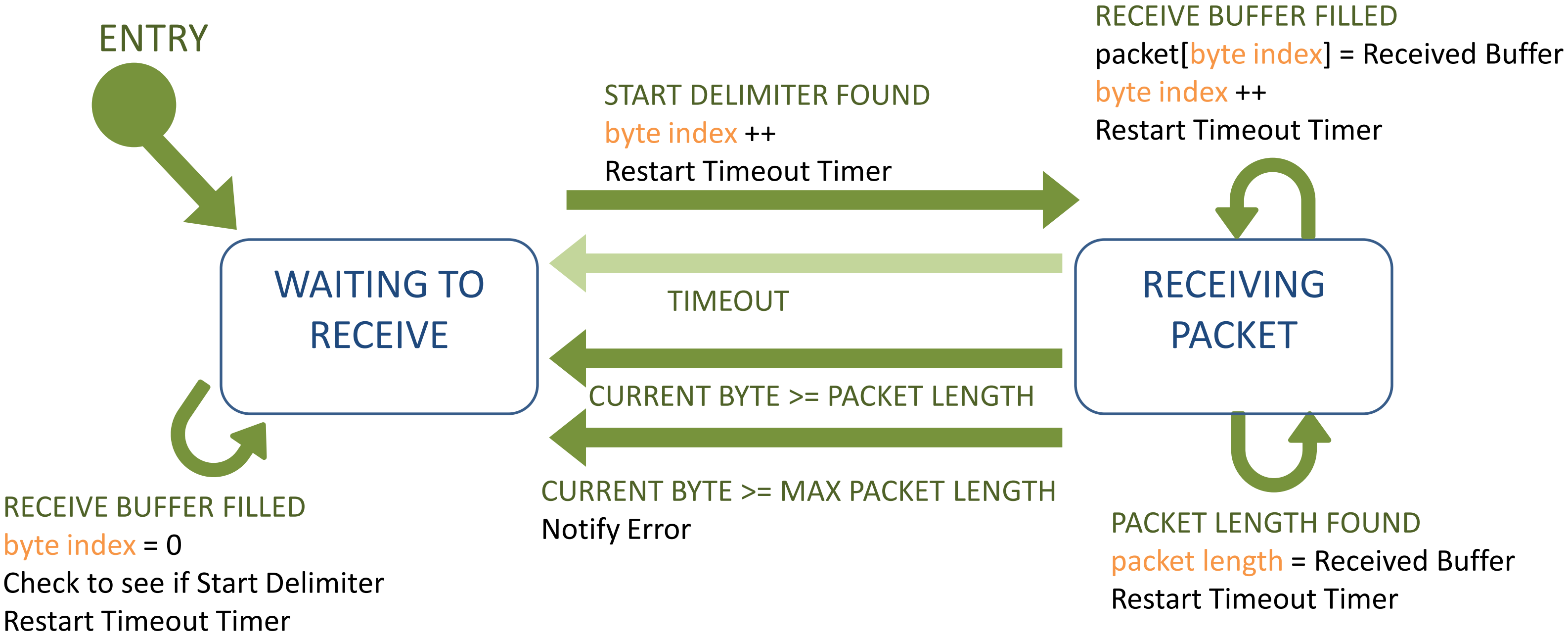
ALWAYS…

NOT IMPLEMENTED

WAITING TO PLAY

TURNING AROUND

SM RECEIVE



ENTRY

WAITING TO RECEIVE

RECEIVE BUFFER FILLED
byte index = 0
Check to see if Start Delimiter
Restart Timeout Timer

START DELIMITER FOUND
byte index ++
Restart Timeout Timer

TIMEOUT

CURRENT BYTE >= PACKET LENGTH

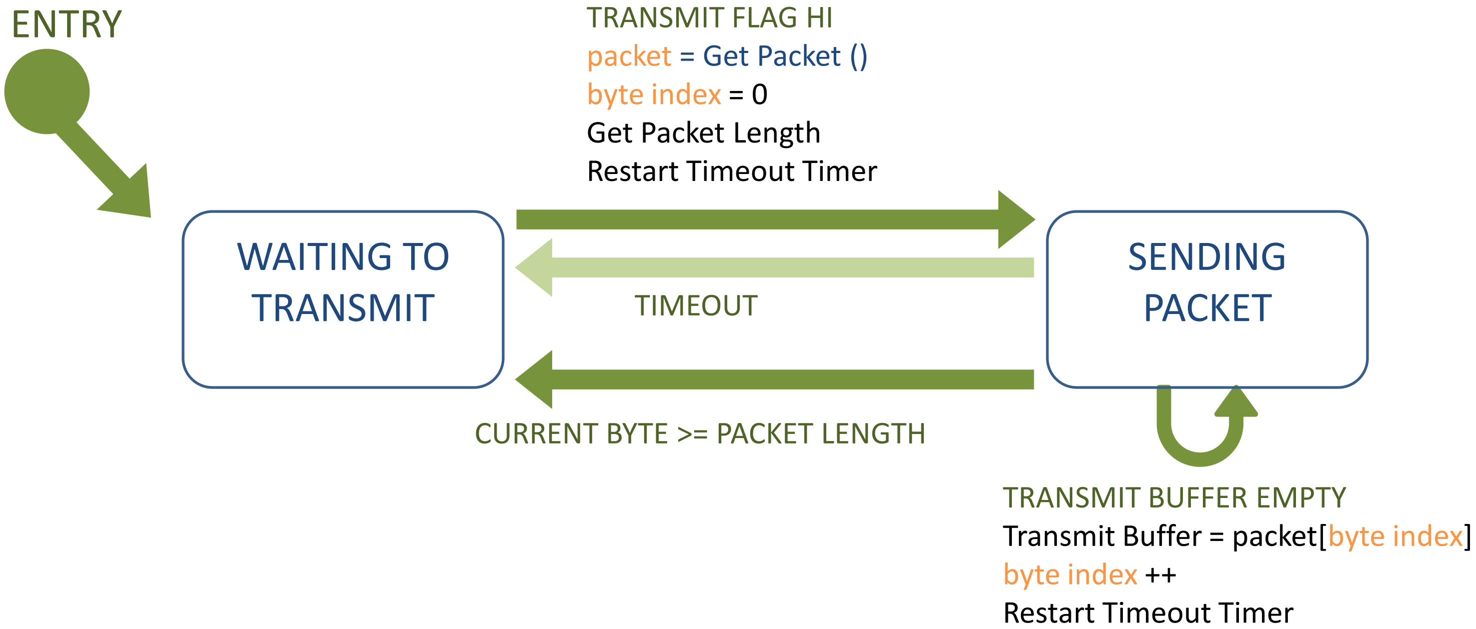
CURRENT BYTE >= MAX PACKET LENGTH
Notify Error

RECEIVING PACKET

RECEIVE BUFFER FILLED
packet[byte index] = Received Buffer
byte index ++
Restart Timeout Timer

PACKET LENGTH FOUND
packet length = Received Buffer
Restart Timeout Timer

SM TRANSMIT



ENTRY

WAITING TO TRANSMIT

SENDING PACKET

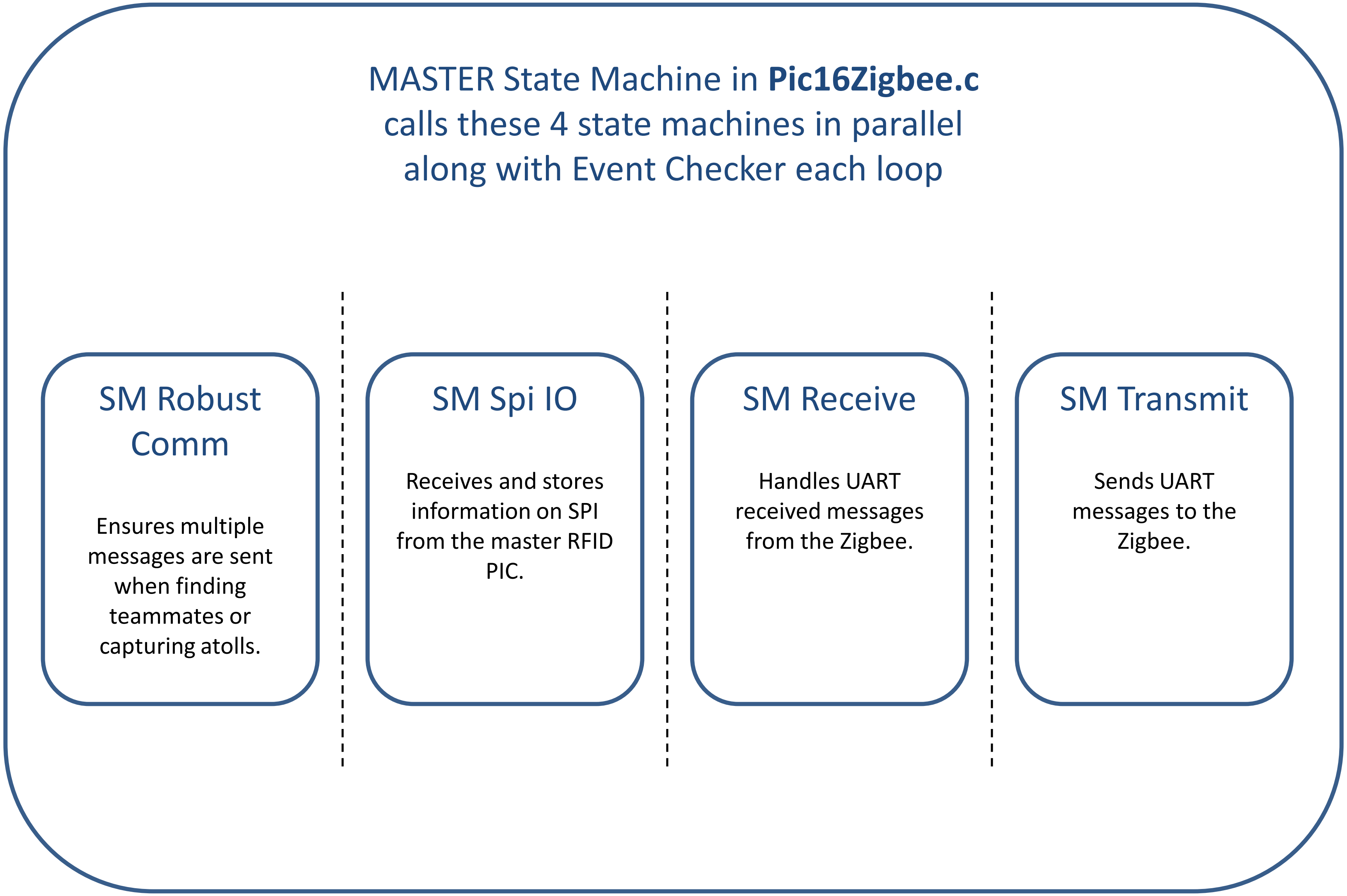
TRANSMIT FLAG HI
packet = Get Packet ()
byte index = 0
Get Packet Length
Restart Timeout Timer

TIMEOUT

CURRENT BYTE >= PACKET LENGTH

TRANSMIT BUFFER EMPTY
Transmit Buffer = packet[byte index]
byte index ++
Restart Timeout Timer

ACV XBee / Communication PIC:
STATE MACHINES



MASTER State Machine in **Pic16Zigbee.c** calls these 4 state machines in parallel along with Event Checker each loop

SM Robust Comm

Ensures multiple messages are sent when finding teammates or capturing atolls.

SM Spi IO

Receives and stores information on SPI from the master RFID PIC.

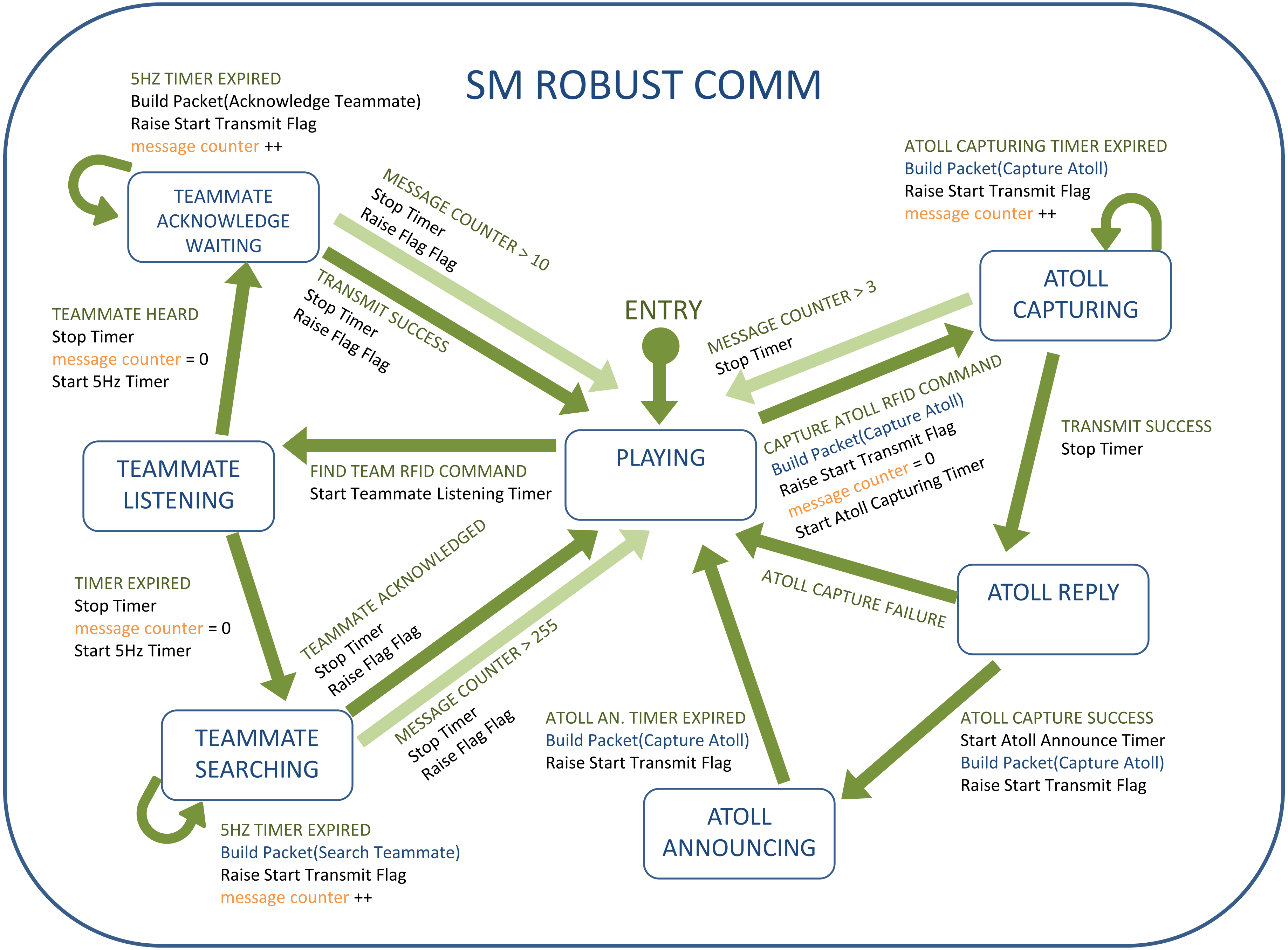
SM Receive

Handles UART received messages from the Zigbee.

SM Transmit

Sends UART messages to the Zigbee.

SM ROBUST COMM



ENTRY

PLAYING

TEAMMATE ACKNOWLEDGE WAITING

TEAMMATE LISTENING

TEAMMATE SEARCHING

ATOLL CAPTURING

ATOLL REPLY

ATOLL ANNOUNCING

5HZ TIMER EXPIRED
Build Packet(Acknowledge Teammate)
Raise Start Transmit Flag
message counter ++

MESSAGE COUNTER > 10
Stop Timer
Raise Flag Flag

TRANSMIT SUCCESS
Stop Timer
Raise Flag Flag

TEAMMATE HEARD
Stop Timer
message counter = 0
Start 5Hz Timer

FIND TEAM RFID COMMAND
Start Teammate Listening Timer

TIMER EXPIRED
Stop Timer
message counter = 0
Start 5Hz Timer

TEAMMATE ACKNOWLEDGED
Stop Timer
Raise Flag Flag

MESSAGE COUNTER > 255
Stop Timer
Raise Flag Flag

5HZ TIMER EXPIRED
Build Packet(Search Teammate)
Raise Start Transmit Flag
message counter ++

ATOLL CAPTURING TIMER EXPIRED
Build Packet(Capture Atoll)
Raise Start Transmit Flag
message counter ++

MESSAGE COUNTER > 3
Stop Timer

CAPTURE ATOLL RFID COMMAND
Build Packet(Capture Atoll)
Raise Start Transmit Flag
message counter = 0
Start Atoll Capturing Timer

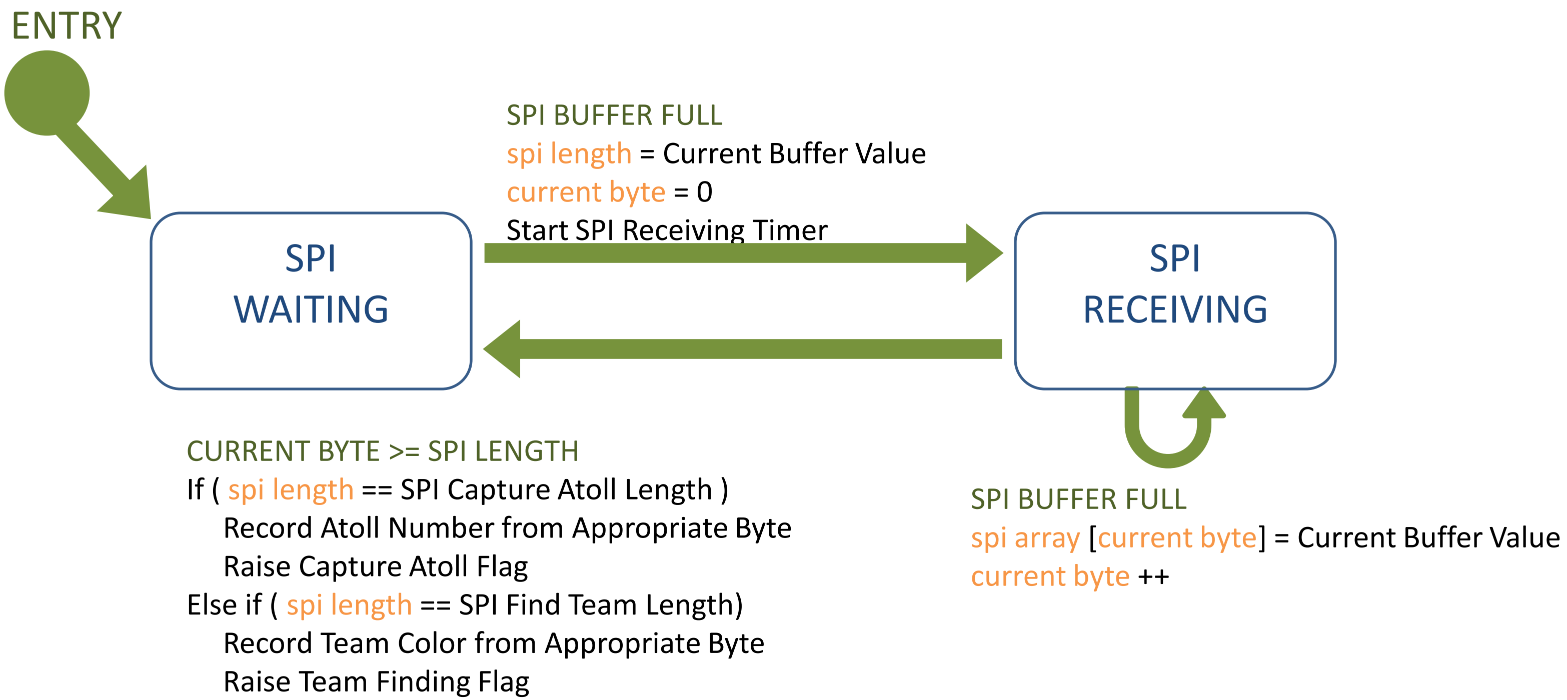
TRANSMIT SUCCESS
Stop Timer

ATOLL CAPTURE FAILURE

ATOLL CAPTURE SUCCESS
Start Atoll Announce Timer
Build Packet(Capture Atoll)
Raise Start Transmit Flag

ATOLL AN. TIMER EXPIRED
Build Packet(Capture Atoll)
Raise Start Transmit Flag

SM SPI IO



ENTRY

SPI WAITING

SPI RECEIVING

SPI BUFFER FULL

```
spi length = Current Buffer Value
current byte = 0
Start SPI Receiving Timer
```

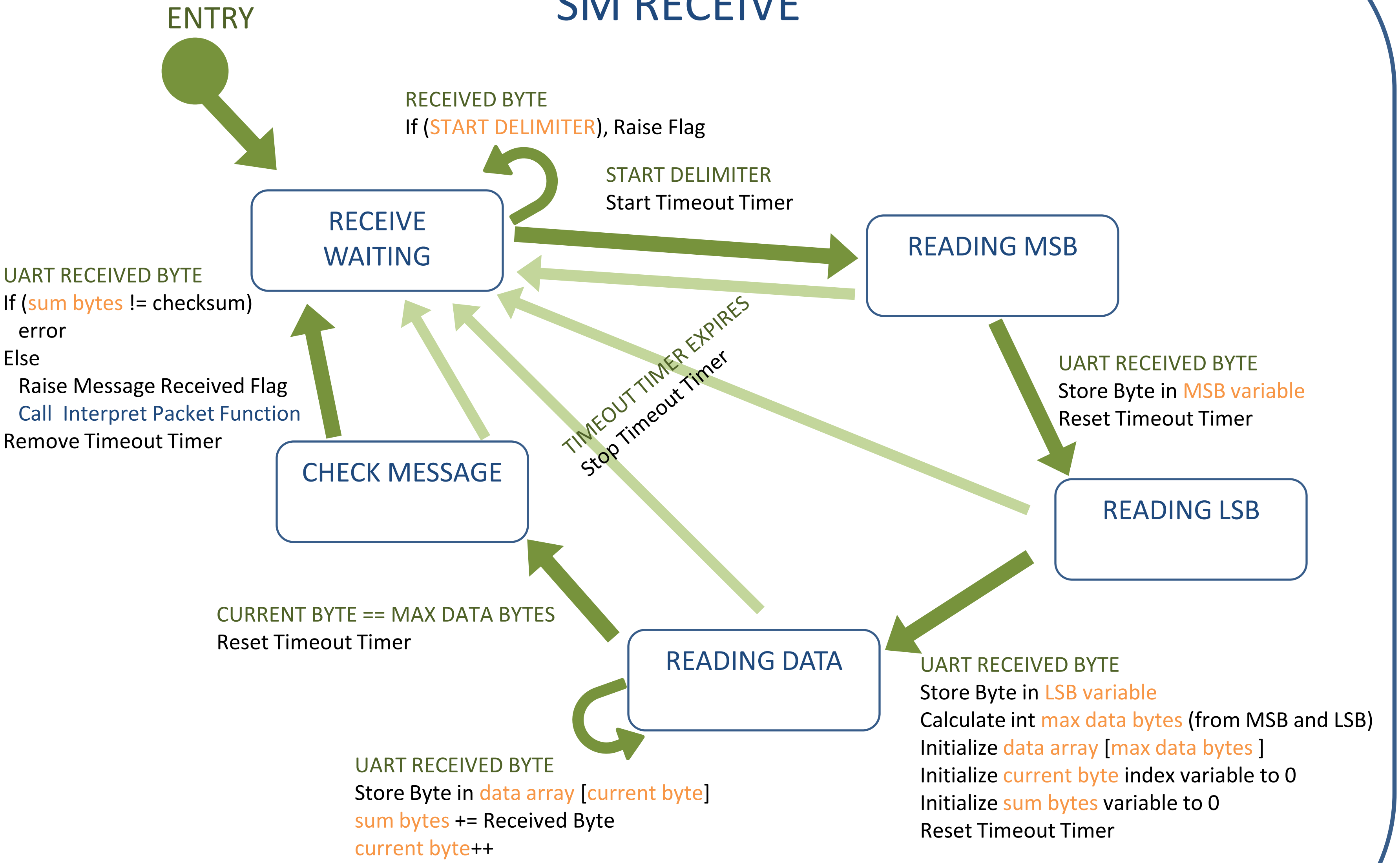
CURRENT BYTE >= SPI LENGTH

```
If ( spi length == SPI Capture Atoll Length )
    Record Atoll Number from Appropriate Byte
    Raise Capture Atoll Flag
Else if ( spi length == SPI Find Team Length)
    Record Team Color from Appropriate Byte
    Raise Team Finding Flag
```

SPI BUFFER FULL

```
spi array [current byte] = Current Buffer Value
current byte ++
```

SM RECEIVE



RECEIVED BYTE
If (START DELIMITER), Raise Flag

START DELIMITER
Start Timeout Timer

UART RECEIVED BYTE
If (sum bytes != checksum)
  error
Else
  Raise Message Received Flag
  Call Interpret Packet Function
Remove Timeout Timer

TIMEOUT TIMER EXPIRES
Stop Timeout Timer

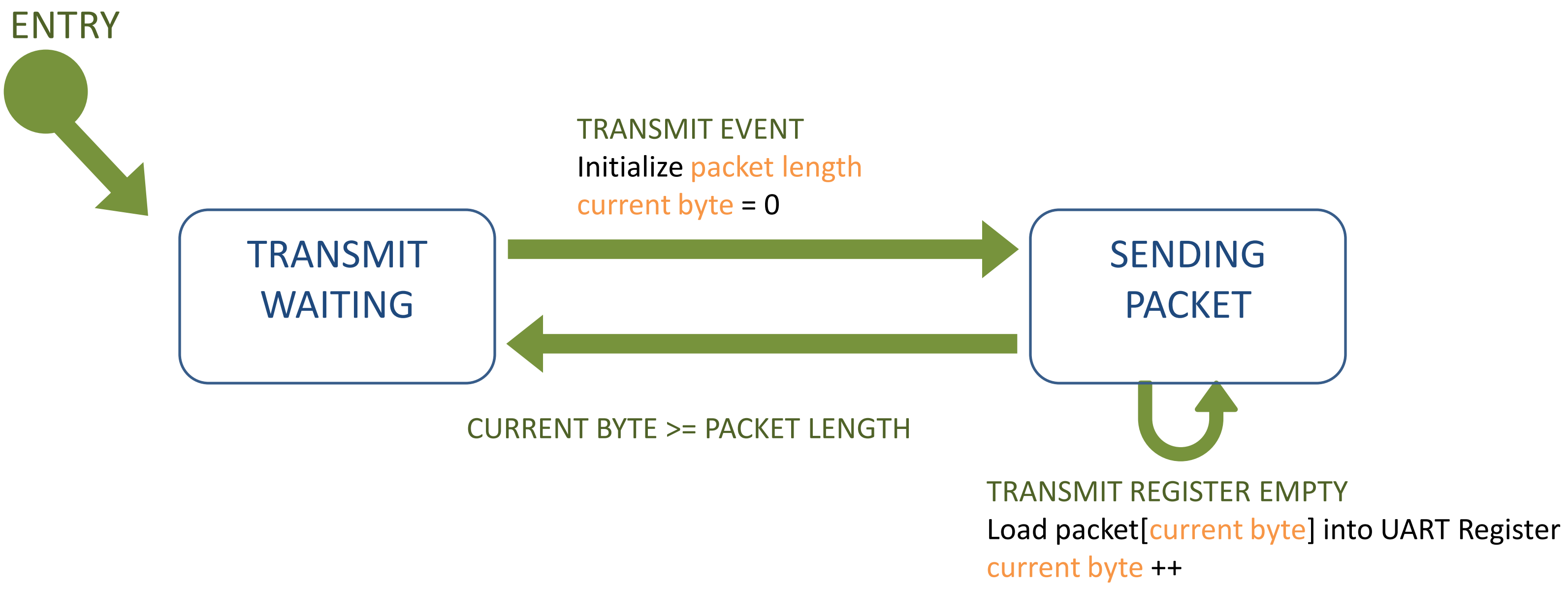
UART RECEIVED BYTE
Store Byte in MSB variable
Reset Timeout Timer

CURRENT BYTE == MAX DATA BYTES
Reset Timeout Timer

UART RECEIVED BYTE
Store Byte in LSB variable
Calculate int max data bytes (from MSB and LSB)
Initialize data array [max data bytes]
Initialize current byte index variable to 0
Initialize sum bytes variable to 0
Reset Timeout Timer

UART RECEIVED BYTE
Store Byte in data array [current byte]
sum bytes += Received Byte
current byte++

SM TRANSMIT



ENTRY

TRANSMIT WAITING

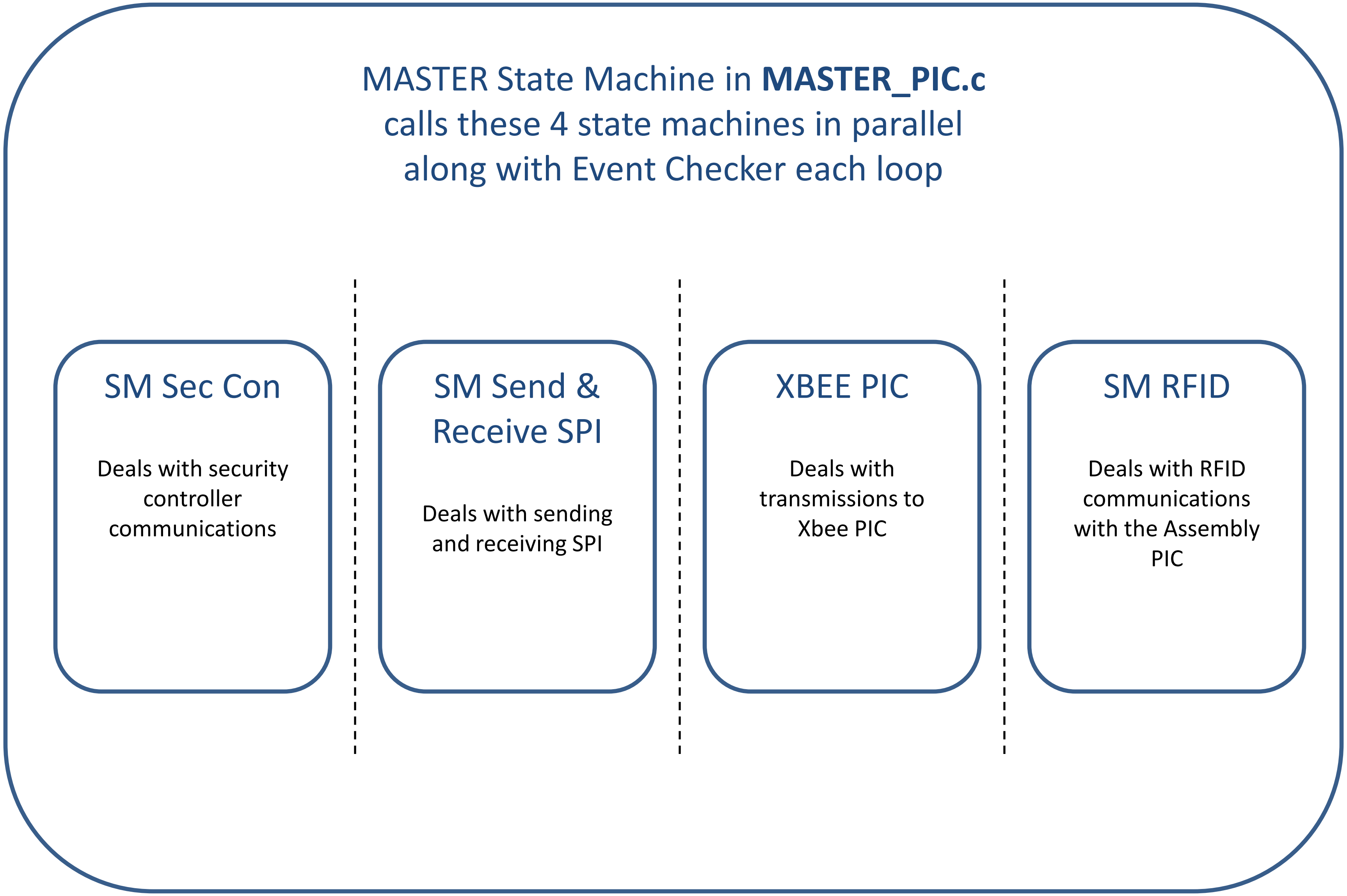
TRANSMIT EVENT
Initialize packet length
current byte = 0

SENDING PACKET

CURRENT BYTE >= PACKET LENGTH

TRANSMIT REGISTER EMPTY
Load packet[current byte] into UART Register
current byte ++

ACV RFID MASTER PIC:
STATE MACHINES



MASTER State Machine in **MASTER_PIC.c** calls these 4 state machines in parallel along with Event Checker each loop

SM Sec Con

Deals with security controller communications

SM Send & Receive SPI

Deals with sending and receiving SPI

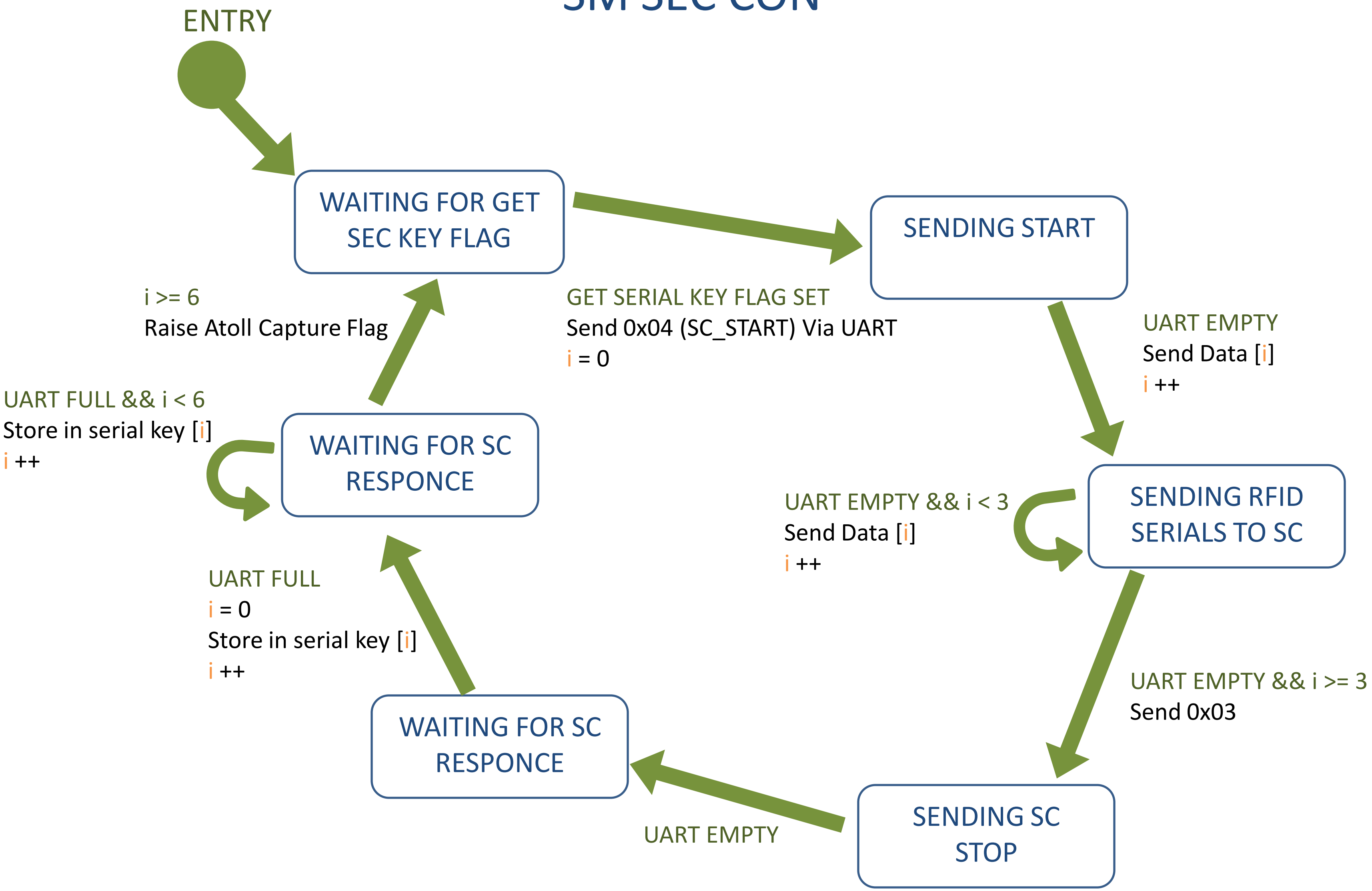
XBEE PIC

Deals with transmissions to Xbee PIC

SM RFID

Deals with RFID communications with the Assembly PIC

SM SEC CON



ENTRY

WAITING FOR GET SEC KEY FLAG

GET SERIAL KEY FLAG SET
Send 0x04 (SC_START) Via UART
i = 0

SENDING START

UART EMPTY
Send Data [i]
i ++

SENDING RFID SERIALS TO SC

UART EMPTY && i < 3
Send Data [i]
i ++

UART EMPTY && i >= 3
Send 0x03

SENDING SC STOP

UART EMPTY

WAITING FOR SC RESPONCE

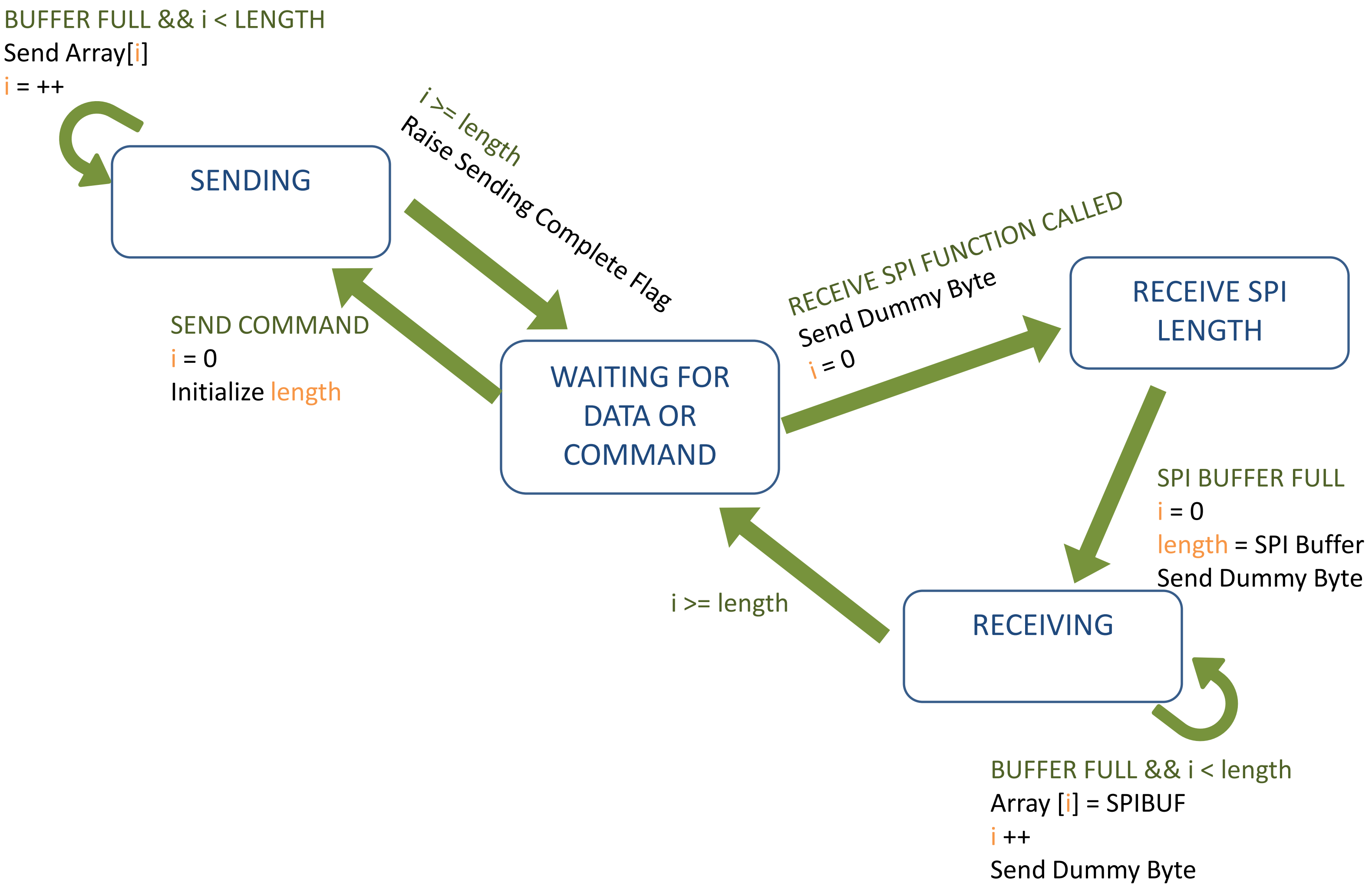
UART FULL
i = 0
Store in serial key [i]
i ++

WAITING FOR SC RESPONCE

UART FULL && i < 6
Store in serial key [i]
i ++

i >= 6
Raise Atoll Capture Flag

SM SEND RECEIVE SPI



BUFFER FULL && i < LENGTH
Send Array[i]
i = ++

SENDING

i >= length
Raise Sending Complete Flag

SEND COMMAND
i = 0
Initialize length

WAITING FOR DATA OR COMMAND

RECEIVE SPI FUNCTION CALLED
Send Dummy Byte
i = 0

RECEIVE SPI LENGTH

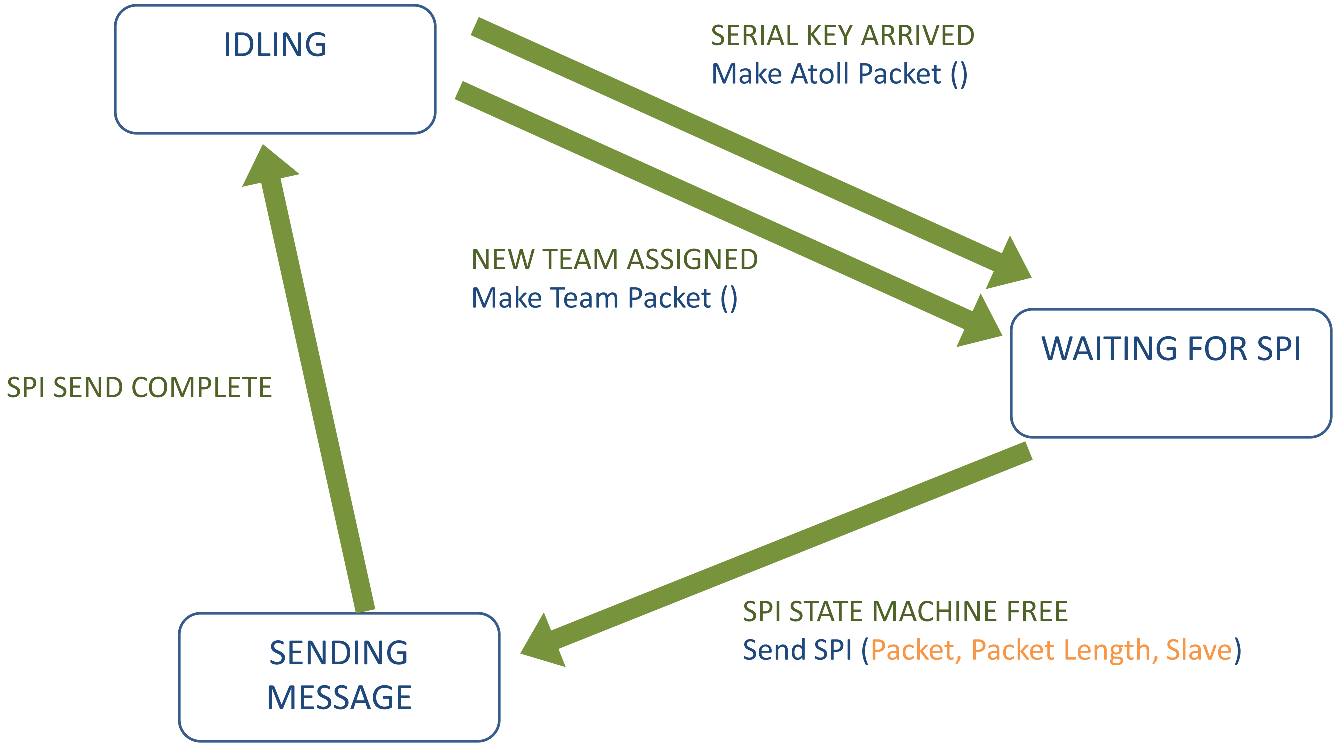
SPI BUFFER FULL
i = 0
length = SPI Buffer
Send Dummy Byte

i >= length

RECEIVING

BUFFER FULL && i < length
Array [i] = SPIBUF
i ++
Send Dummy Byte

SM XBEE



IDLING

SERIAL KEY ARRIVED
Make Atoll Packet ()

NEW TEAM ASSIGNED
Make Team Packet ()

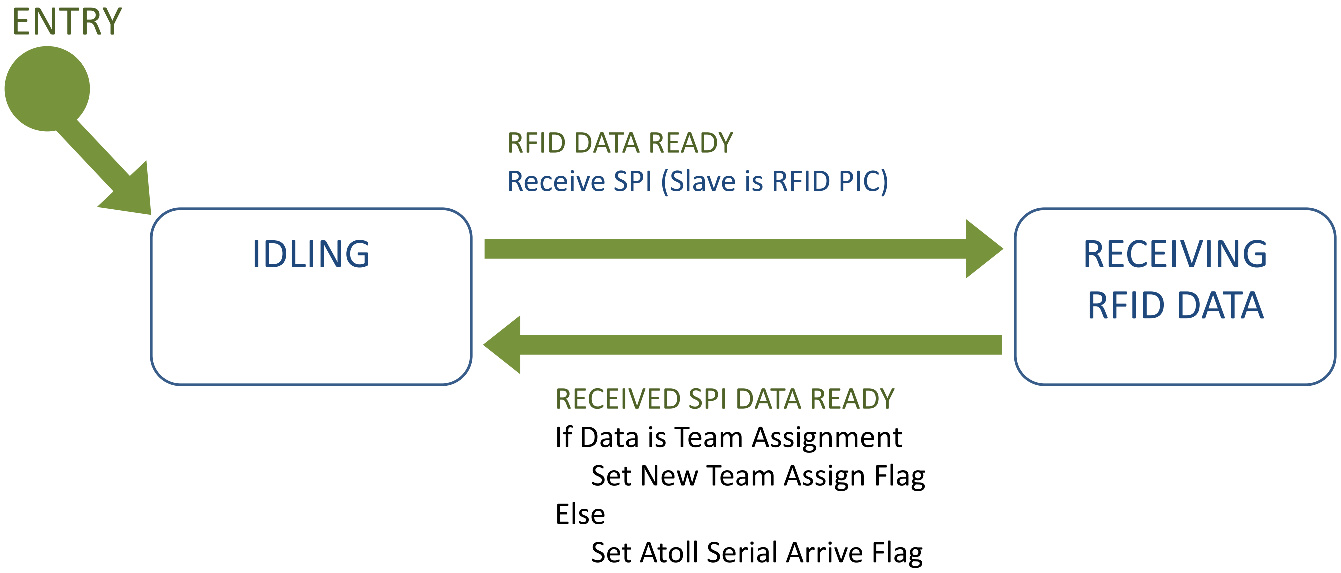
WAITING FOR SPI

SPI SEND COMPLETE

SENDING MESSAGE

SPI STATE MACHINE FREE
Send SPI (Packet, Packet Length, Slave)

SM RFID



ENTRY

IDLING

RFID DATA READY

Receive SPI (Slave is RFID PIC)

RECEIVING RFID DATA

RECEIVED SPI DATA READY

```
If Data is Team Assignment
    Set New Team Assign Flag
Else
    Set Atoll Serial Arrive Flag
```