**LTTO Flow chart**

LTTO::begin (txPin, rxPin) LTTO.cpp

Constructor.

Sets up Tx and Rx pins.

Sets up Timer0 mid cycle 1mS interrupt for ISR

Calls SetupPinChangeInterrupt()

SetupPinChangeInterrupt() ISR.cpp

Calls enableInterupt() to set up a Pinchange interrupt on the Rx pin.

Adds a pointer to the Instance in isrArray[] – using RxPin as the index

ISRchange() ISR.cpp

Triggered when there is a level change on any pin.

Checks which pin triggered the interrupt

Calls PinChange()

PinChange() ISR.cpp

Measures Marks/Breaks

Stores the information in messageIR[]

SIGNAL (Timer0) ISR.cpp

Measures Marks/Breaks

Decrements receiveMilliTimer

Calls CreateIRmessage() 9mS after the last pinChange interrupt

LTTO::CreateIRmessage() ReceiveIR.cpp

Measures Marks/Breaks

Decodes the message type ( T, B, P, D, C ) and stores it in \_messageIRtype

Decodes the message byte and stores in \_messageIRdataPacket

Calls PushToFifo()

LTTO::PushToFifo() FIFO.cpp

Stores all incoming messages into the FIFO

FIFO size is declared in LTTO.h (const byte FIFO\_SIZE = 20;)

LTTO::PopFromFifo() FIFO.cpp

Recalls the information from the FIFO

Populates it into decodedIRmessage.type

decodedIRmessage.rawDataPacket

Sets decodedIRmessage.type = ‘ ‘ to show it is empty (null)

LTTO::available ReceiveIR.cpp

Called by the Main Loop

If not called often enough the FIFO will overflow and explode

Cleans up the mess when an overflow occurs.

Calls PopFromFifo() to recall the most recent datapacket

Uses decodedIRmessage.type to call;

* ProcessTag
* ProcessBeacon
* ProcessPacket
* ProcessDataByte
* ProcessCheckSum

Sets decodedIRmessage.newMessage = false

LTTO::ProcessTag ProcessIR.cpp

Calculates decodedIRmessage.teamID

Calculates decodedIRmessage.playerID

Calculates decodedIRmessage.shotStrength

LTTO::ProcessBeacon ProcessIR.cpp

x

LTTO::ProcessPacket ProcessIR.cpp

Sets decodedIRmessage.checkSumCalc = 0

Sets decodedIRmessage. byteCount = 0

Sets decodedIRmessage.checkSumOK = false

Adds decodedIRmessage.packetByte to CheckSum

Calculates decodedIRmessage.packetByte

Calculates decodedIRmessage.packetName

LTTO;:ProcessDataByte ProcessIR.cpp

Increments decodedIRmessage. byteCount

Calculates decodedIRmessage.dataByte

Adds decodedIRmessage.packetByte to CheckSum

LTTO:;ProcessCheckSum ProcessIR.cpp

Compares decodedIRmessage.checkSumRxByte to CheckSumCalc

Sets decodedIRmessage.checkSumOK = true/false

-------------------------------------------------------------------------------------------------------------------------

**LTTO:Hosting**

State = HOST calls HostMode()

HostMode()

Set # of Teams

TeamTags

SlowTags

Start sets state = ANNOUNCE GAME

State= ANOUNCE\_GAME calls AnnounceGame()

AnnounceGame()

Select Team of next player

Calls AdvertiseGame() every 1500mS

AssignPlayer listens for JoinRequests and calls AddPlayer

Listens for AckPlayerAssignment

Adds Tagger to database