```
*****************************************************************
```

Pseudo-code for Master SM (Top Level)

```
Module variables: MyPriority, CurrentState, MyTeam
States: WAITING_TO_START, CONSTRUCTING
Events Posted: ES_GAME_START (StartGameQuery EventChecker in comm),
ES_GAME_OVER (one-shot game timer interrupt posts)
```

InitMasterSM

```
Takes a priority number, returns True.

Initialize the MyPriority variable with the passed in parameter
Set ThisEvent Type to ES_ENTRY
Decide what team we are on:
     Read the input port of the TEAM_PIN
     Set MyTeam variable to the team we are on
     Make necessary changes (??) - its going to be a flag so later
     we can use if statements that determine what to do depending on
     team color
Call InitAll (initialize hardware, initialize all interrupts,
initialize PWM)
Call StartMasterSM Function with ES_ENTRY

End of InitMasterSM (return True)
```

RunMasterSM

```
Takes ES_Event CurrentEvent, returns ES_NO_EVENT

Set MakeTransition variable to false, because we are not making a
transition currently
Set state type variable NextState to CurrentState
Set event type EntryEventKind to ES_ENTRY (default to normal entry to
new state)
Set event type ReturnEvent to ES_NO_EVENT, assuming no error

Switch (CurrentState)
     Case WAITING_TO_START
          Execute During function for WAITING_TO_START
          If the event is active (not ES_NO_EVENT)
               Switch (Event)
                    Case: ES_GAME_START
                         Start Game Timer (One shot timer for 2:18)
                         by calling StartGamerTimer function
```

```
                        Set NextState to CONSTRUCTING
                        Set MakeTransition to true
                        Set ReturnEvent to ES_NO_EVENT
                    End Case
                End Switch
            End if
        End Case

        Case CONSTRUCTING
            Call the DuringConstructing function
            Set CurrentEvent to returned event from during function
            If the CurrentEvent is active (not ES_NO_EVENT)
                Switch (CurrentEvent)
                    Case ES_GAME_OVER
                        Stop Motors
                        Set next state to WAITING_TO_START
                        Set MakeTransition to true
                        Set ReturnEvent to ES_NO_EVENT (consumed)
                    End Case
                End Switch
            End if
        End Case

        If MakeTransition is true (we are transitioning to a different
        state)
            Set the CurrentEvent to ES_EXIT
            Call RunMasterSM with CurrentEvent
            Set CurrentState to NextState
            Call RunMasterSM with ES_ENTRY event (start the entry
            function for the new state)
        Endif
    Return ReturnEvent
    End RunMasterSM
```

StartMasterSM

```
Takes ES_EVENT Current Event, returns nothing

Initialize CurrentState to WAITING_TO_START
Call RunMasterSM with Current Event (ES_ENTRY event)
```

DuringWaitingToStart

```
Takes Event, returns Event
```

```
Do nothing
Return Event
```

DuringConstructing

Takes event, returns event

```
If event is ES_ENTRY or ES_ENTRY_HISTORY
    Start the constructing state machine by calling
    StartConstructingSM
Else if event is ES_EXIT
    If exiting constructing state, give the lower levels a chance
    To clean up first
    Call RunConstructingSM
Else
    Run any lower level state machine
    Call RunConstructingSM
Endif
Return Event (this event is either an event that MasterSM needs to
handle, or ES_NO_EVENT if a lower level SM handled it)
```

<u>Public Functions</u>

GetTeam

Returns MyTeam

<u>Private Functions</u>

StartGameTimer

Takes nothing, returns nothing

```
Start timer and enable stall in debugging
```

GameTimerInterruptResponse

Takes nothing, returns nothing

```
Clear the source of the interrupt (one-shot timer)
Post ES_GAME_OVER event to MasterSM
End GameTimerInterruptResponse
```