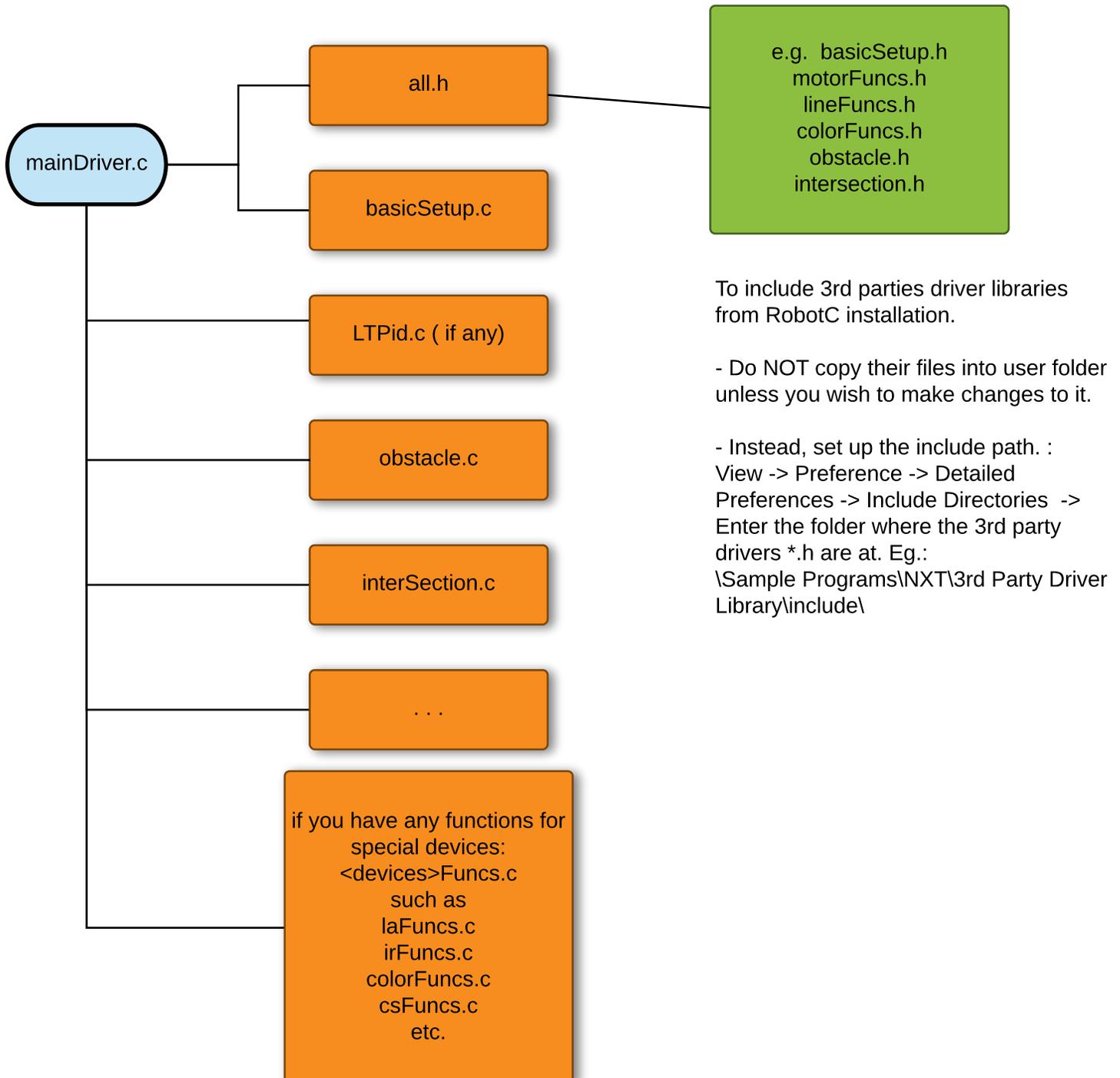


Typical Files System for RCJ Rescue Line Project

Sample Final folder should contain...

Make sure your filename reflect its own sub-task.



Sample Layout...

```
// the main driver

// all the *.c files have already included proper *.h files

#include "lt.c"
#include "intersection.c"
#include "obstacle.c"
// etc.
task main()
{
    call setup / initialization function
    // this setup function is the 1st thing to call in any test tasks
    // before doing anything else.

    while ( not evac room)
    {
        if (see obstacle)
            doObstacle ()
        if (see intersection)
            doIntersection ()
        if (see gap)
            doGap ()
        // etc.
        doLineTracing ()
    }

    call evac func ()
}
```

```
// The all.h

#ifndef __ALL_H__
#define __ALL_H__

... global variables for everyone,
such as
const tmotor LeftMotor = motorA
... etc.

// if your members have different chassis

// do NOT create different files just because there are a
few parameters are different.
// rule of thumbs: this situation should be minimized also

#ifdef _Member1Name_
const float wheelBase = 20.5;
// here put in all other globals that contain different value
among members.
#elif _Member2Name_
const float wheelBase = 17.25;
#else _Member3Name_
const float wheelBase = 17.45;
#endif

#endif
```

```
// the intersection h

#ifndef __INTERSECTION_H__
#define __INTERSECTION_H__

... global variables specific just to intersections

#endif
```

```
// the intersection.c

#include "all.h"
#include "intersection.h"

all functions pertaining just to intersection
```

When you are working on your own sub-task

Assuming this is your team's
current files system

```
— doc
— final
— member1
    intersection.c
    intersection.h
    testIntersection.c
— member2
    obstacle.c
    obstacle.h
    testObstacle.c
— stage
    all.h
    obstacle.h
```

Assuming member 1 wants to use the all.h in "stage"

```
// the intersection.c

#include "../stage/all.h"
#include "intersection.h"

all functions pertaining just to intersection
```

Before you copy your code to stage...

```
// the testIntersection.c
```

```
#include "intersection.c"
```

```
task main() {
    initialization()

    while (1) {
        if (see intersection)
            doIntersection()
    }
}
```

Notice : there is NO task main()

```
// the intersection.c version for integration test and
// copied into "stage" folder
```

```
#include "all.h" // note : not "../stage/all.h"
```

```
#include "intersection.h"
```

```
bool seeIntersection() {
    ...
}
```

```
void doIntersection() {
    ...
}
```

NOTES:

- MUST update their status often
- MUST follow update protocol among members, such as:
 - ▶ what should be in stage
 - ▶ no one should modify the files in stage
 - ▶ unit tests must be done inside each member's own folder.
 - ▶ etc.
- MUST have a gatekeeper for the main shared Libray functions and header, such as mainLib.c and all.hSet up google hangout among teammates
- setup Googlesync