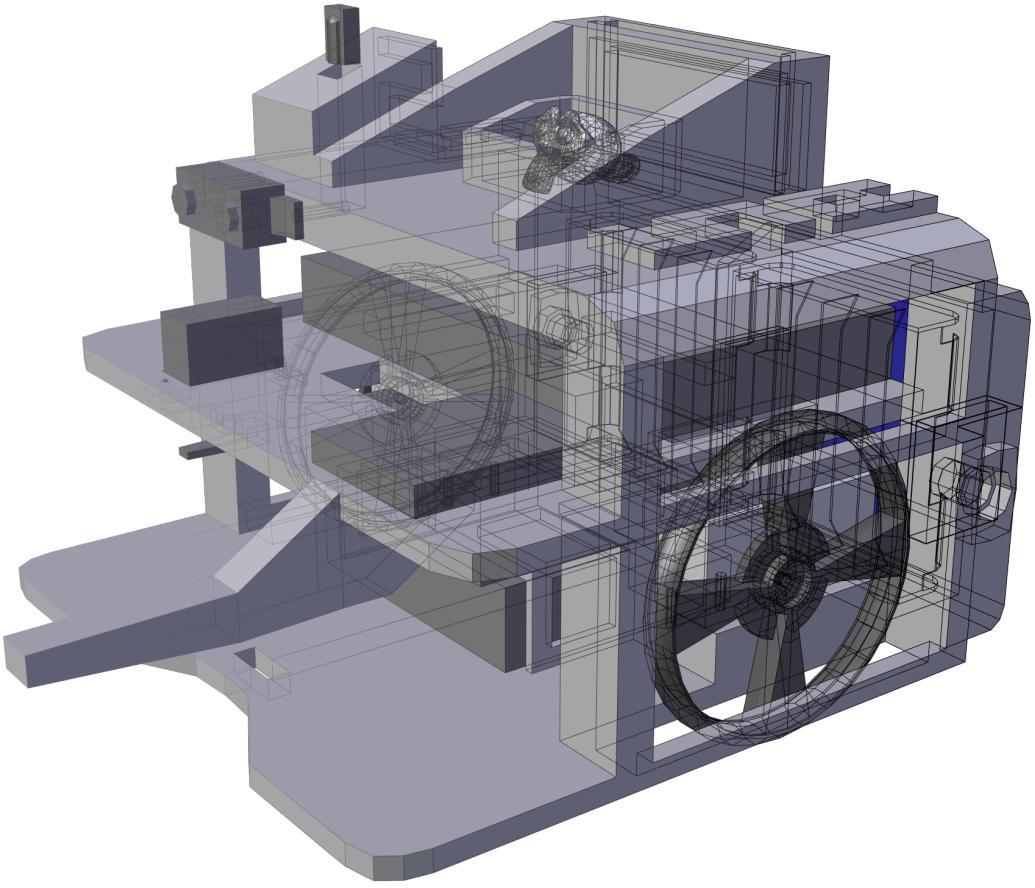
The Capture The Flag Project

Welcome to the capture-the-flag(CTF) project. Throughout the duration of this project you will be striving to create a CTF bot capable of navigating a ring, capturing a flag and returning said flag back to its original base. There are several factors that differentiate this project from the fire fighter project but the main differences are the capture the flag bot's use of asynchronous serial communications. This project is not extremely complex but allows for more creative ideas when compared to the firefighter bot. Overall, this project will allow students to explore and test their computer engineering skills to new degrees than previously explored in the TEJ3UI course.



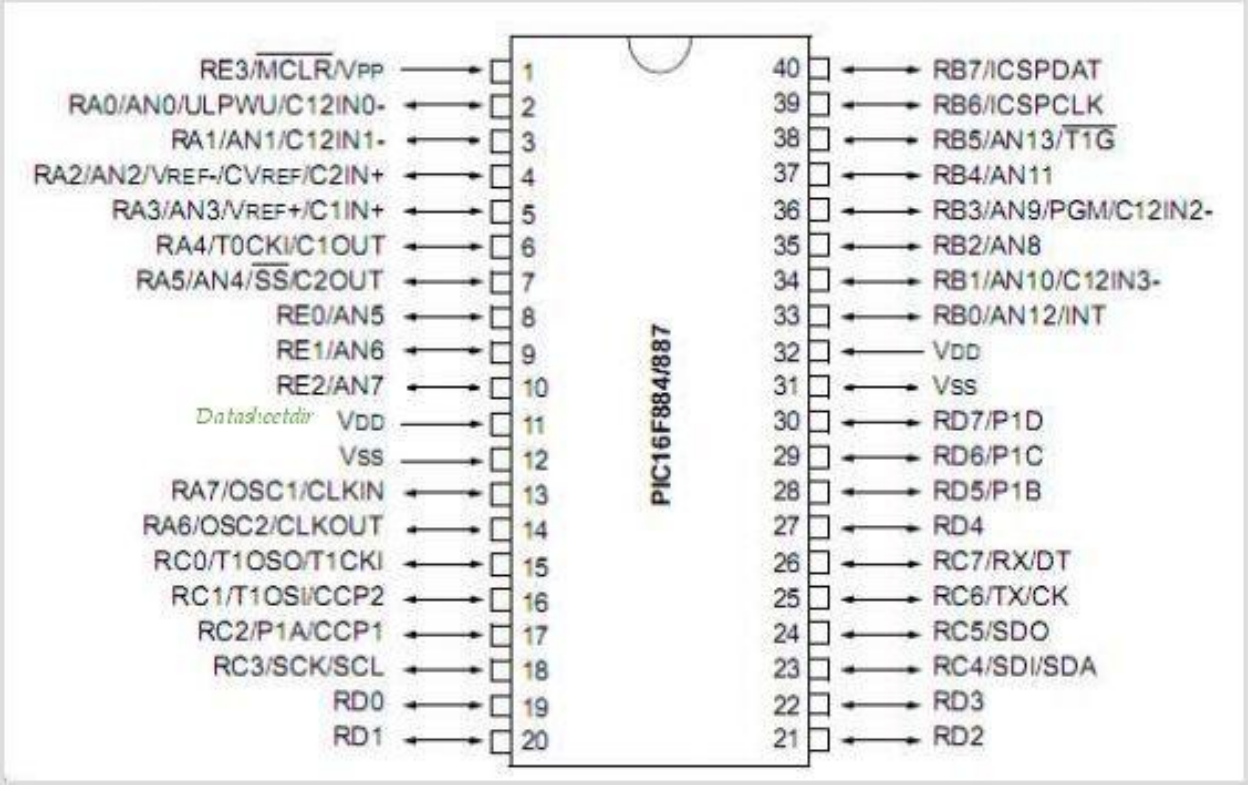
Serial Communications

What Is Serial Communications?

Sometimes in circuitry, it is necessary for chips to be able to communicate wirelessly. This can be done through serial communications. For the purpose of the capture-the-flag project, the type of serial communications you will be needing is asynchronous serial communications.

How Do We Use It?

Asynchronous serial communications is a method of data transfer that involves sending data one bit at a time at a set clock rate. In asynchronous serial communications, the object that is sending out data will be sending data at a defined rate and the object receiving said data must know what this defined rate is in order to receive it. This is the type of serial communications that you will be using between your bot and the flag. For the purpose of this project, your flag will be constantly outputting data, at a rate of 1200 bits per second, which your bot will receive in order to identify that the object in question is a flag. For serial communications, there are two pins on the PIC16F887 that we must use.



Datasheetdir

These are pins 26 and 25, the RX and TX pins. The PIC16F887 is capable of sending and receiving data through asynchronous serial communications. The RX pin is the receiving pin and the TX pin is the transmitting pin.

Through two labs, you will learn the basics of serial communications. The first lab will be wired for the data transfer method, and the second will be wireless. This will teach you how to program and use serial communications for your bots.

While this covers the basics of what the chip uses for serial communications, there are two important commands that allow you to use these functions.

The Commands

SERIN

SERIN *Pin*,*Mode*,{*Timeout*,*Label*,}{**[***Qual...***]**,}{*Item...*}

The serin command receives serial data on a specified pin, the RX pin (see PIC16F887 image above), and stores it into a variable.
The various items in the command are shown above.

- Pin refers to a specific pin.
- Mode refers to the baudrate(measure of data bits transferred in one second) of the serial message.
- Timeout is the time that the program waits before going to label, usually indicating an error has occurred.
- Qualification refers to the parity or identifier bit of a serial message.
- The item is the variable that the serial data will be stored into.

Example:

INCLUDE 'modedefs.bas' ← This command is similar to import from python. It calls upon a library where serial commands are and includes that basic file into this one so you can use those commands.

MAIN:

SERIN C6, T1200, 1000, ERROR, ["A"], GauravIsCool '1200 baud rate (bdps)

```
ERROR:                                  ‘ subroutine called when timeout period elapses
LCDOUT $FE, 1, “Sagar is even cooler”
Pause 100
```

SEROUT

SEROUT *Pin*,*Mode*,[*Item*{,*Item...*}]

Similar to the SERIN command, the SEROUT command is used for serial communications. Instead of receiving data, this command allows the pin to output data through its TX pin/pins. The item here refers

Example:

```
True Var Word
True = 0
MAIN:
True = True + 1
SEROUT C7, N1200, [“A”], #True  ‘inverted baud rate of 1200 bdps.
                                ‘ this serial command is sending the decimal value of the variable true.

GOTO MAIN
```

In this program, it is interesting to note that the baud rate was changed. It’s important to note that you can change the baud rate as you please but the same baud rate has to be used on the receiving end as well. For example, if the flag were outputting at a baud rate of 9600 bdps, you would have to define that specific baud rate in your SERIN command.

Marking Scheme

DISCLAIMER:
This marking scheme explains the various checkpoints that you can meet in order to get specific marks. Keep in mind that Mr. Webb will be judging you based largely on your effort as well so don't expect these marks to be final, for better or for worse.

MARKS:	REQUIREMENT:
80	Getting into the middle of the ring. This involves navigating the pit and getting into the center of the ring.
85	Getting into the middle and capturing an object in the middle of the ring.
90	Getting into the middle and capturing the correct object using serial communications.*
90	Getting into the middle, capturing an object and going back to your original starting point. (No serial communications needed)
95	Capturing the correct object and returning to your original starting point.
100	Having the fastest time of capturing the correct object and returning to your original starting point.

***For this, there will be two objects in the middle of the ring. One object will be serially outputting data and the other will not. You will be required to display on your LCD, "Flag Detected", when you receive serial data from the flag. You will then be required to capture the correct object.**

Note: Think of these marks as minimum marks. This means that even if your CTF bot completely broke after you got into the middle, you will still get an 80 at the least.

Rules:

- **No hard-code for getting into the middle as your sole commands. This means fixed motor statements such as : going forward for two seconds, then turning to the right for one second and then going forward for three seconds. Motor controlled routines are allowed, but cannot be your only method of getting into the middle.**

- **Your bot must fit within the Sumo Bot box (8in x 8in x 8in)**

- **Your bot must not pose a danger to any other bots or people in or around the ring**

What Is The Capture The Flag Project?

The Goal:

Each group will compete to produce a working capture the flag robot capable of navigating the ring, identifying and capturing a flag. To do this, you will need some previous concepts such as line detection along with some new concepts such as use of a sharp sensor for detecting distance of objects and use of asynchronous serial communications. If your bot is the fastest to capture the flag and return to its home base, then it will be declared the winner.

Obstacles:

The Ring:

When your bot gets in the ring, it will be placed at one of the four bases around the outside of the ring. From the outside you will have to navigate around the pit and get into the center of the ring so that you can capture the flag. Getting around this pit can be challenging but keep in mind that the pit is outlined by white paint so you can detect the outside edges using your line detection.

The Flag

Another obstacle is the flag itself. One of the requirements to get a higher mark on your bot is to be able to identify the flag using serial communications. The flag will be constantly outputting a serial message with an identifier of [“F’] and data of 52. Your bot should be able to differentiate between the flag and a normal object that is not outputting anything such as a chair.

Getting Back to Base

Once you have captured the flag, you must return to your original starting point. You can do this by various methods. You could make a base that your bot would detect or your bot’s program could make sure that the bot retraces its steps getting into the ring. If you are able to capture the flag and go back to your original starting point you can guarantee yourself a 90 or 95 depending on whether or not you used serial communications to capture the “correct” flag.

