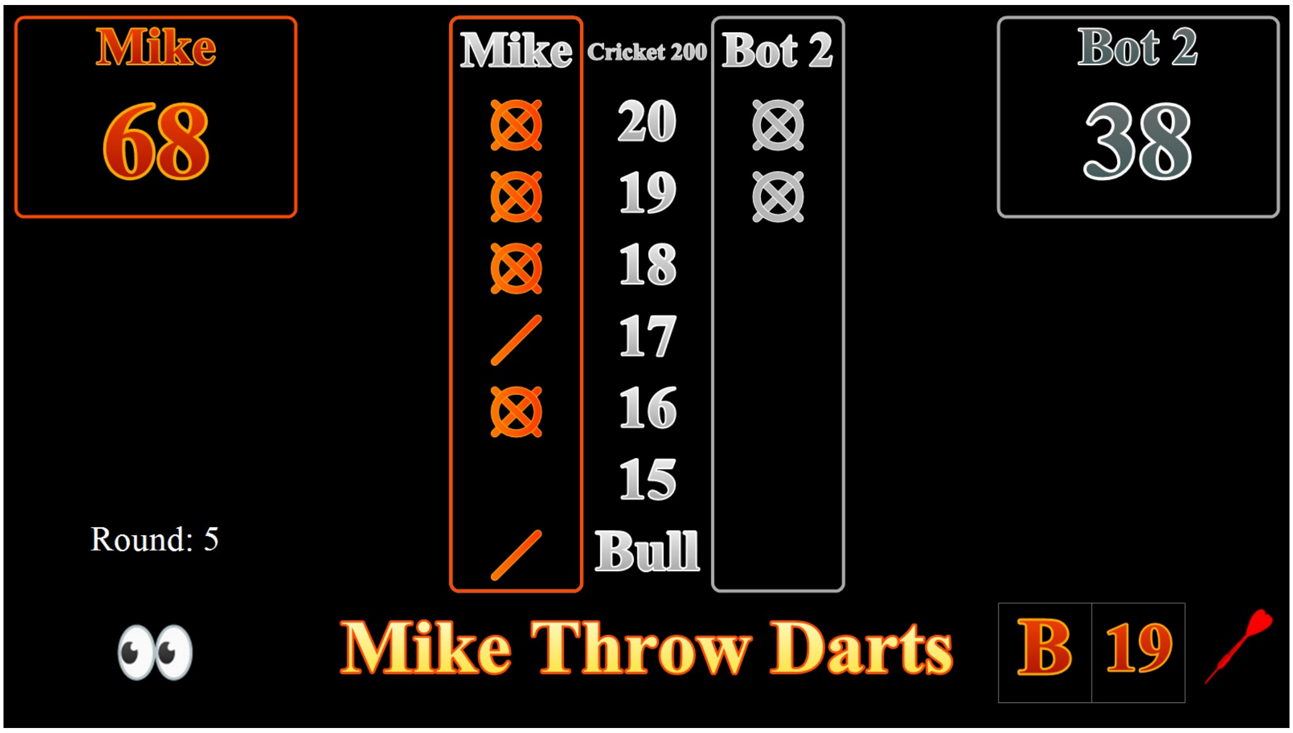
Dart Panel User Manual



OverView

DartPanel is a PC-based application that interacts with electronic dart boards. DartPanel plays the most common dart games such as Cricket and 01, along with a few novelty games. Inspiration for this project was taken from the OpenDarts project by Ricardo Alves (thank you Ricardo!). I personally was unable to load the OpenDarts app on my PC, so I decided to create my own version. I also added a few small additions to the Arduino code, such as a player change LED and button. The Arduino sketch is available along with instructions on how to modify it for your particular board. Analyzing the dart board matrix and wiring the Arduino will likely be the most challenging part of the whole project.

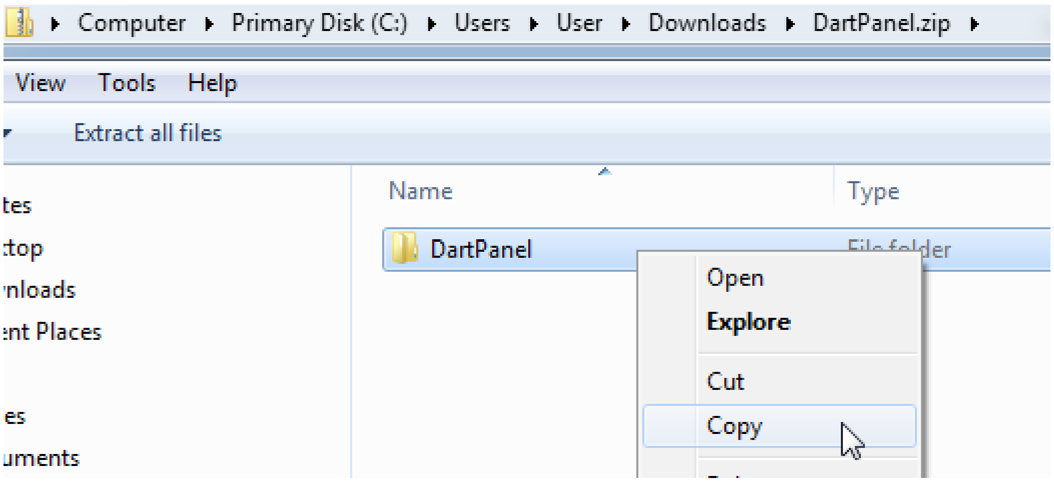
Theory of Operation

An electronic dart board is gutted of its original electronics, and the matrix wires are tapped to detect dart hits. The matrix operates just like a typical PC keyboard. An Arduino is wired to the matrix, and scans the lines continually- looking for hits. When a hit is detected, the line upon which the hit is detected is sent to the PC via serial over the USB connection. The app running on the PC (Dart Panel) then decodes those hit codes and translates them into their corresponding dart board number. The hit is then applied to the current game being played.

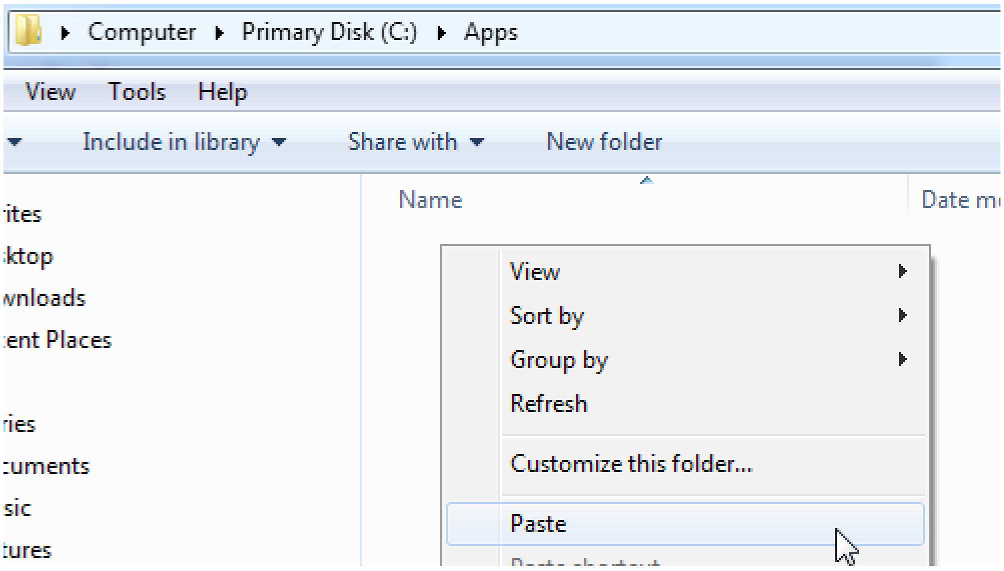
DartPanel Application Download

The DartPanel.exe, support files, and Arduino code are all contained in a zip file found here: http://northosoft.com/download/DartPanel.zip

Download the DatPanel.zip to your local PC and UNZIP the contenets to a directory on your PC, such as C:\Apps\DartPanel. On the typical Windows PC, you may double-click into the zip file, then right-click and select Copy.

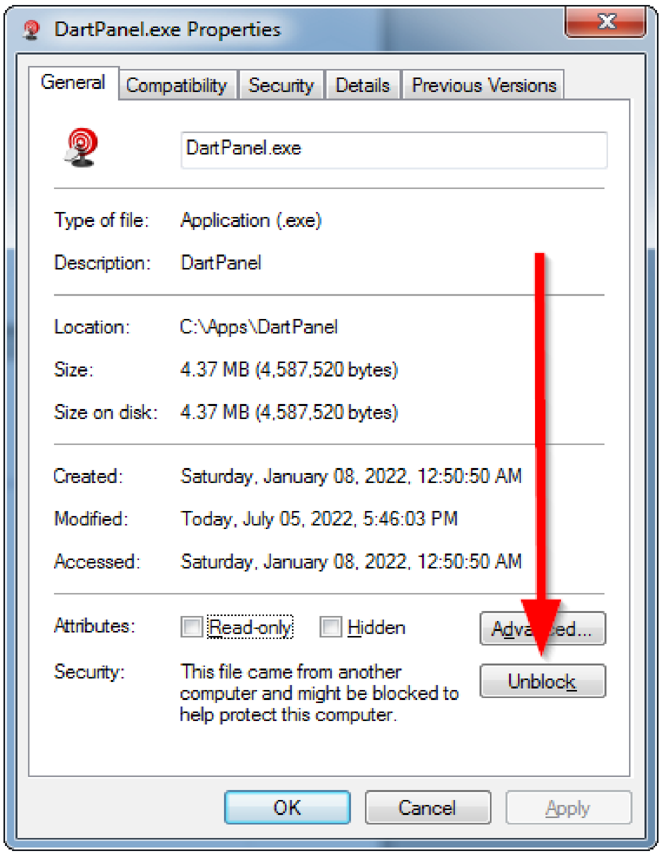

Computer ▸ Primary Disk (C:) ▸ Users ▸ User ▸ Downloads ▸ DartPanel.zip ▸
View Tools Help
Extract all files
Name
Type
DartPanel
Open
Explore
Cut
Copy

You may then Paste the DartPanel folder to your local c drive(or wherever you want to put it)



Windows PC's have become increasingly security-conscious lately, so to help keep this app from popping warning windows, you may right-click the DartPanel.exe, and select "Properties".

From there, click the "Unblock" button.



DartPanel.exe

Type of file: Application (.exe)

Description: DartPanel

Location: C:\Apps\DartPanel

Size: 4.37 MB (4,587,520 bytes)

Size on disk: 4.37 MB (4,587,520 bytes)

Created: Saturday, January 08, 2022, 12:50:50 AM

Modified: Today, July 05, 2022, 5:46:03 PM

Accessed: Saturday, January 08, 2022, 12:50:50 AM

Attributes: Read-only Hidden

Security: This file came from another computer and might be blocked to help protect this computer.

Then click the OK button.

You may now create a shortcut on your desktop to the DartPanel.exe.

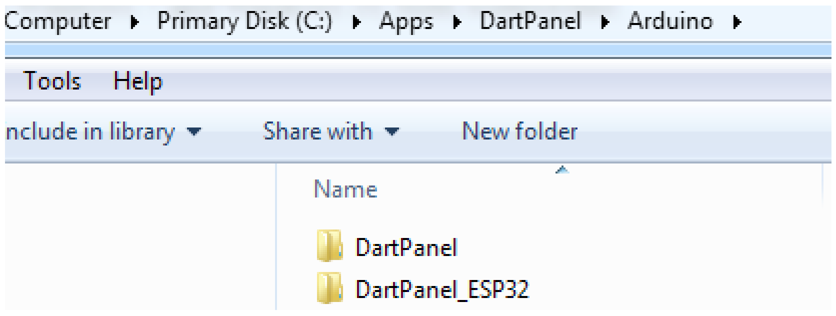
I have tested this Application with Windows 7 and Windows 10, both worked fine.

The PC application is not currently open source, but if you have any suggestions or bugs to report, feel free to send an email to info@northosoft.com

The Arduino folder contains 2 subfolders:

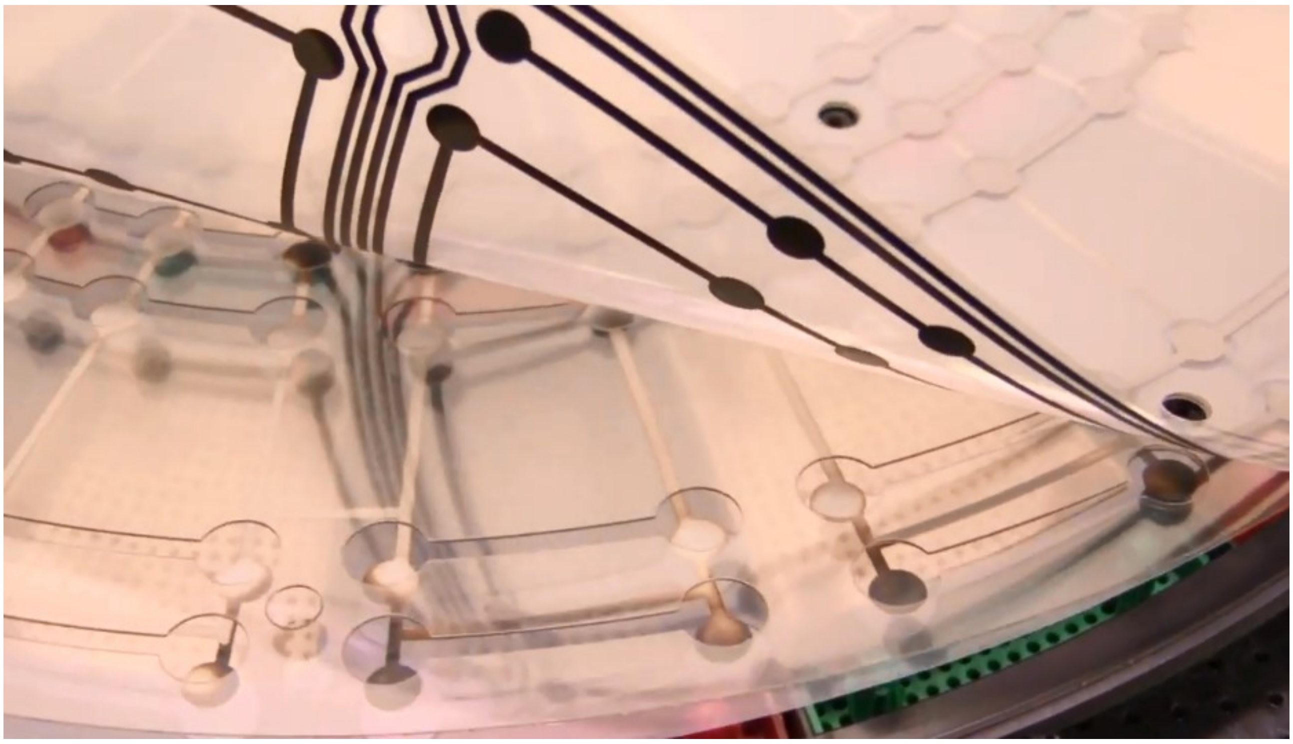
DartPanel: This is the Arduino code as written for an Uno

DartPanel_ESP32: This is the Arduino code as written for the ESP32

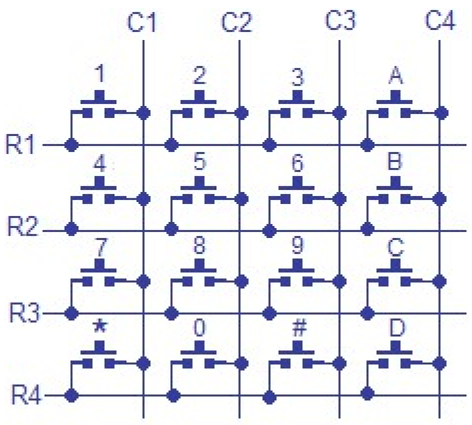


Board Variants

As with other projects of this type (Google Arduino Dart Board Matrix), an important concept that must be understood first is: how the dart board matrix works, and how the wires/leads are scanned. A dart board has 20 numbers, each of which has 3 possible marks (single, double, triple). The bull and double bull add 2 more "keys" to check for. So, this equals (20 x 3 + 2) or 62 unique keys to scan for. Using a matrix, this means that ideally 16 wires would be used (8 x 8 = 64). The matrix has 2 layers sandwiched together with dots for each segment that act like momentary switches.



Here is a simple conceptual diagram of how a switch matrix works:



This example matrix has 4 rows and 4 columns, so the output connector has 8 total pins. When you read in rows and columns, this allows for 16 unique values to be read.

The dart board matrix works the same way. It may look like a bunch of spaghetti, but the principal is the same.

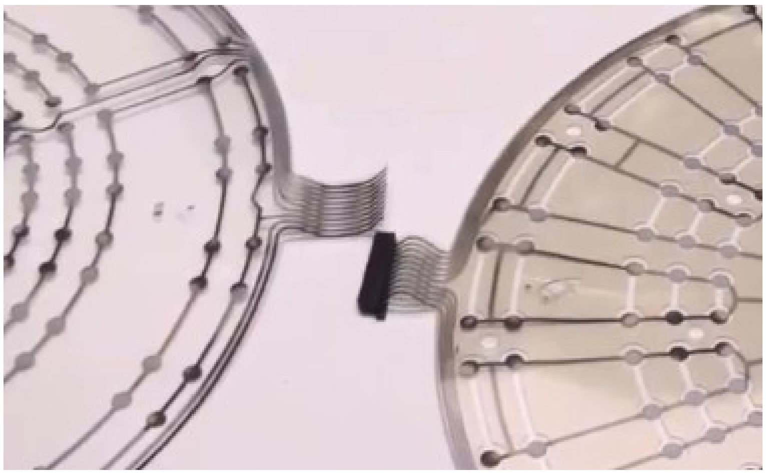
Donor Boards

As of the writing of this document, I have personally dismantled 4 different dart boards. I like the Arachnid boards over other brands I have seen, as it seems like their construction and matrix quality is superioir.

1) Arachnid English Mark Darts board, 301 only. Circa 1990's. No double bull, but the matrix supported it. I modified with a replacement double bull, but it was a real pain and required a mini-lathe. This board is very well made, and had 3 flat cables coming from the matrix. I ended up reusing the circuit board from this one, and cut the original CPU out. I dremeled out traces that did not involve the dart head, and tapped into the 16 pins that fed the original CPU. It worked, but was quite the rats nest of wires. I found this board in the local classifieds for $20, as the owner had lost the power adapter.
2) Arachnid Cricket Pro 650, currently available, but originally came out in the early 2000's. I found this board in the local classifieds for $20, as the owner had lost the power adapter- seeing a pattern here? Matrix cables were two 11 pin flat flex cables. The bottom one had 3 unused lines, so the 8 lines were the Master lines. The top was the 11 slave lines.
3) Arachnid Cricket Pro 800, currently available. This one I just peeked in, because I remember paying big bucks for it new. The matrix cables were the same as the ones on the 650.
4) Unicorn Cheapie. This one I got at a thrift store and left it at work (previous job). The board quality was very low, and I would probably not waste my time modifying such a cheap board.

Whichever board you start with, the first thing to figure out is what the matrix arrangement is. The two Arduino examples in the /Arduino subfolder will handle 8x8 or 8x11. If this is a new to you, the following YouTube video gives a great tour of the guts of a typical dart board.
https://www.youtube.com/watch?v=OqGiXrJ_FI8

Each layer of the matrix will have one or two ribbon cables.

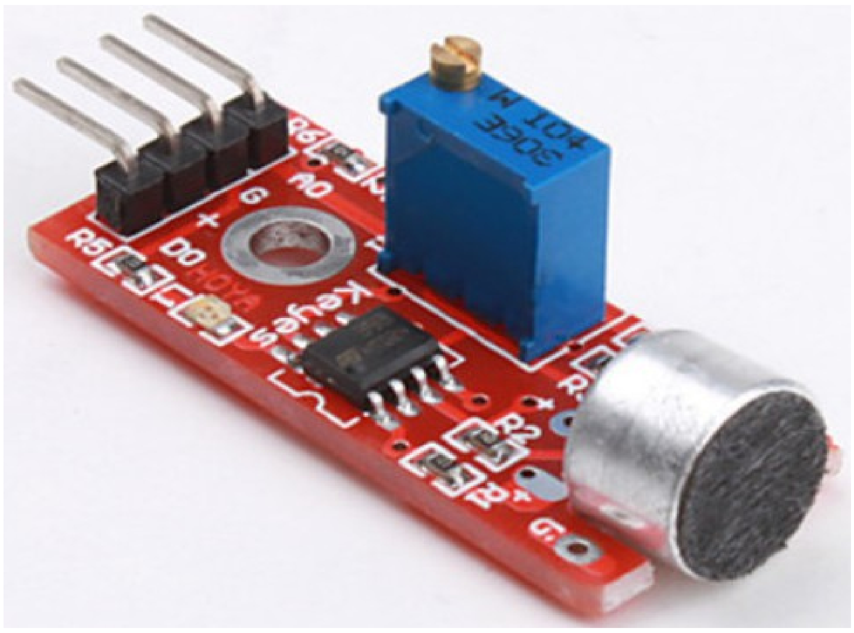


These cables are called FPC (Flexible Printed Cable) or sometimes FFC (Flat Flexible Cable). The connectors are often proprietrary and/or difficult to source.

Miss Sensor

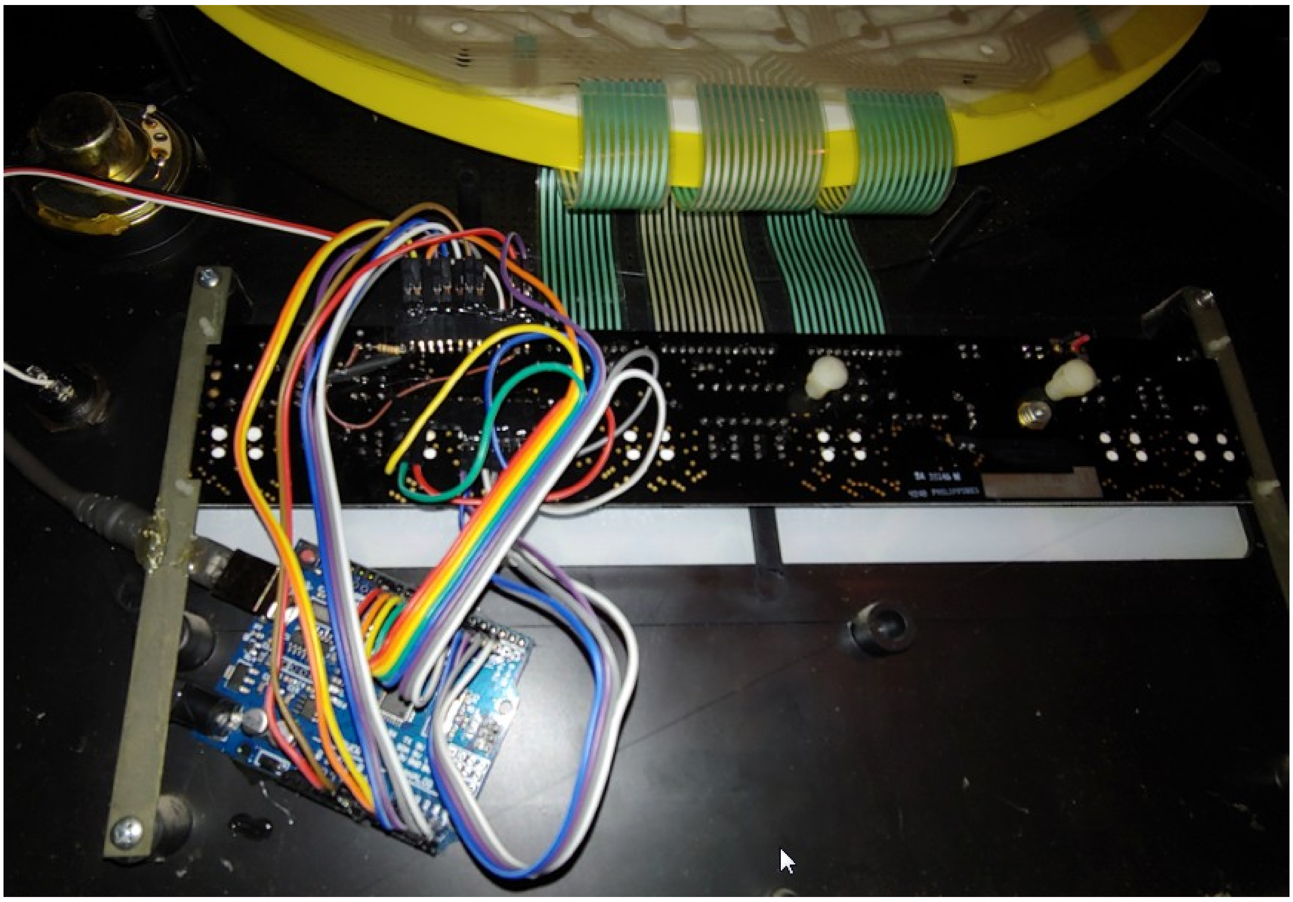
I used a cheap sound/microphone detection module as a missed dart sensor. This module outputs a quick TTL Low pulse when it detects a sound. There is also a sensitivity pot. These cost around $2 and work pretty well, so that is what I used. If you don't want to use one, no big deal- just don't. You can edit the Arduino code to comment out the miss sensor stuff if you want to.

On the board where I used the Arduino Uno, I actually ran out of pins. I ended up sharing the miss sensor input with the player change button on the same pin. Since the pulse out of the sensor is very quick, I timed the duration of the detected Low signal. If is was < 50MS, I consider that a miss. Else, I consider that a player change. So far it seems to work OK.



Miss Sensor

The first board I converted was the Arachnid English Mark Darts 301 board. This one was a little convoluted, so I won't go into a ton of detail.



This board matrix had 3 flex cables running into the original circuit board. The different thing about this arrangement is that some of the lines were electrically tied together. This reduces the actual number of lines to scan. I was able to clip the legs of the original CPU, and trace the 16 lines/pins required to scan the matrix. I identified the master lines by observing the 8 lines tied to pull up resistors. I actually ran a wire from the 5v of the Arduino to the common line of this resistor array. I figured, if the original board used them, I might as well too.

After wiring up the array, I uploaded the DartPanel.ino sketch (found in the \Arduino folder).

I used the Arduino Serial Monitor to observe dart hits.

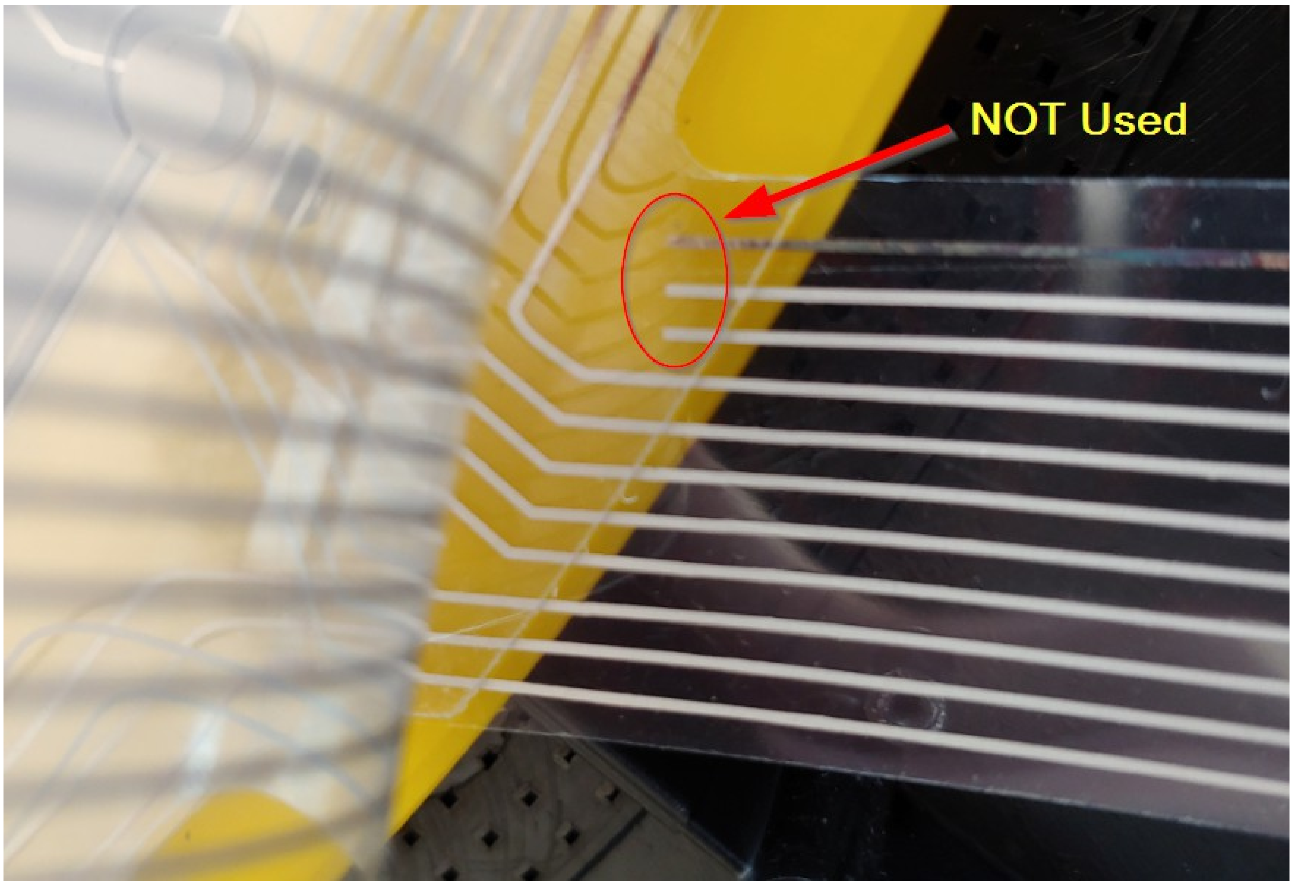
Note- this sketch requires the Arduino to receive a "C" character to "Connect", so that the Arduino knows that it has been connected to by a legitimate application. At boot up, the player change LED will blink rapidly so you know that the board is not connected. You can do this by entering C at the top of the Serial Monitor.

Next, the board needs to be changed into "Shooting State" by sending an "S" character. Again, you can also do this by entering S and sending (or press enter) on the Serial Monitor.

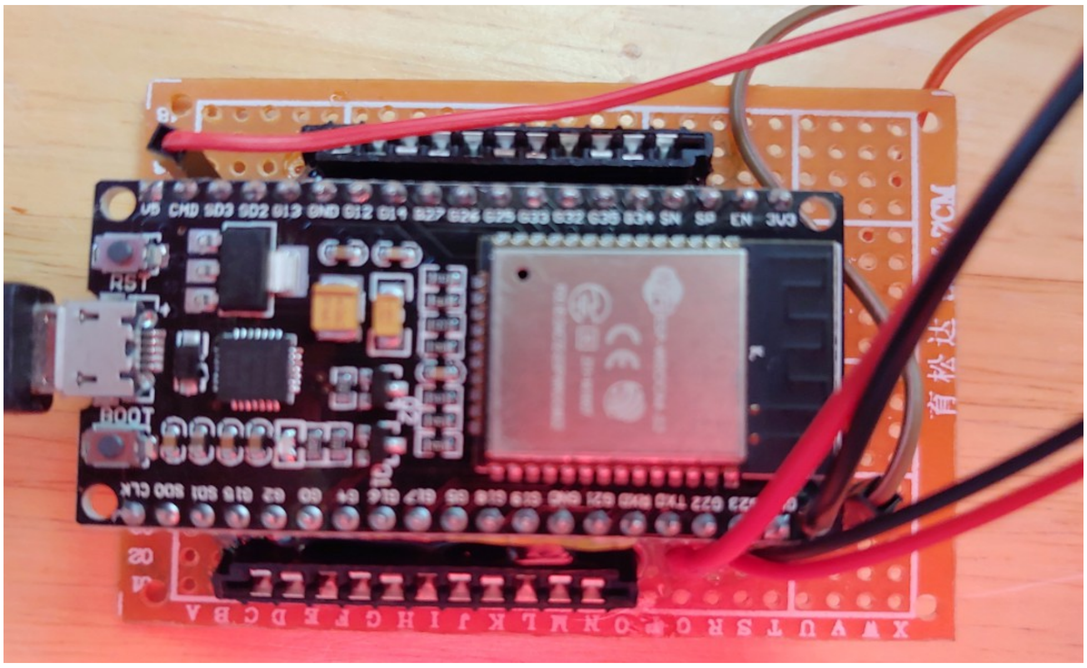
On the original board, I manually (tediously) pressed each number on the board and recorded each code into a file called piemap.txt. The DartPanel.exe app has since been modified to make this process much easier (more on that later).

Arachnid 650 Build

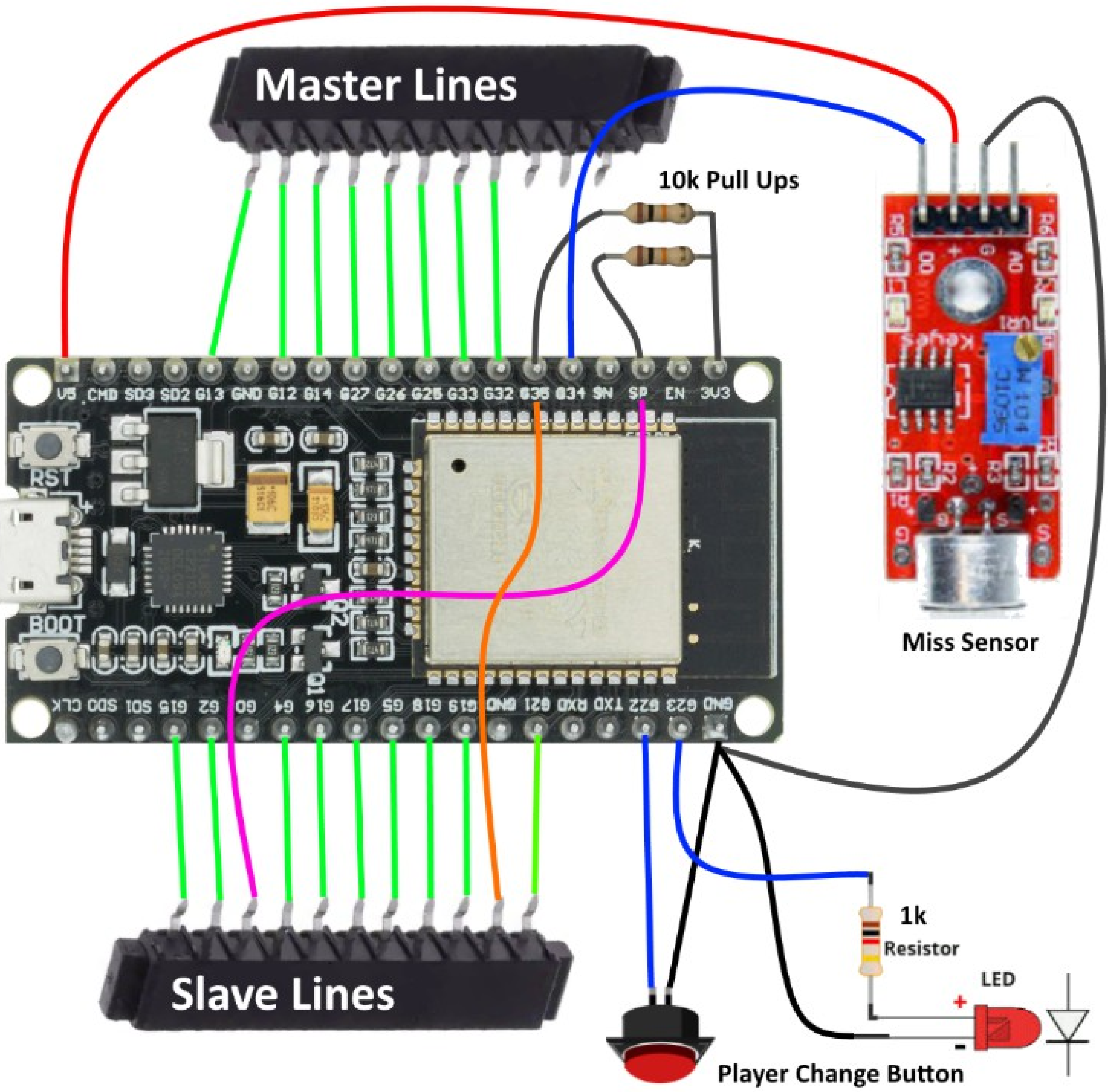
This one was much "cleaner" than the first board, so for that reason I will go into more detail on this one. This board has a matrix with two 11 line flat flex cables. On the lower cable, 3 of the 11 lines were not used. This is what I used for the "Master Lines"



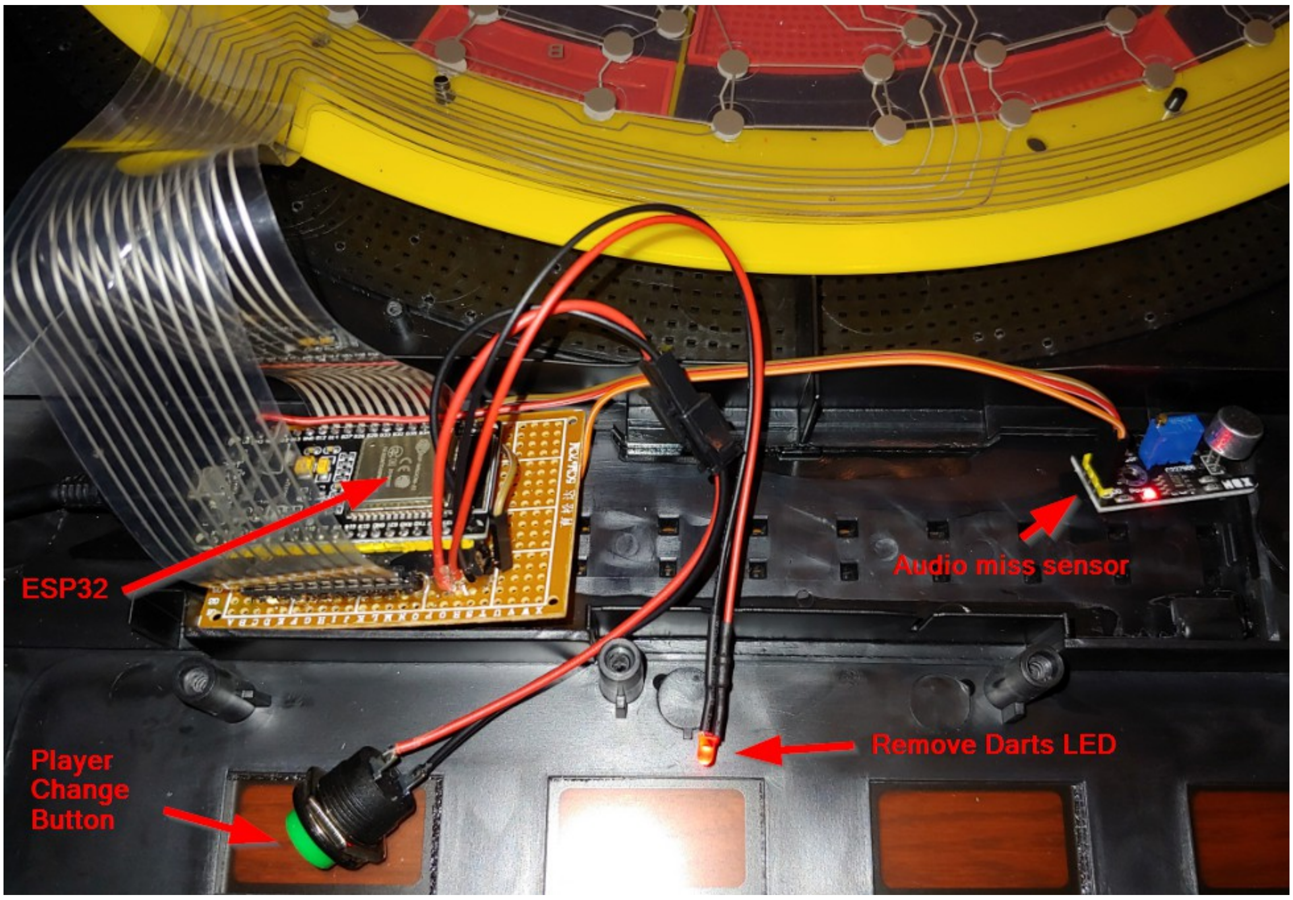
I had an esp32 in the parts box, so this is what I used for the microcontroller.



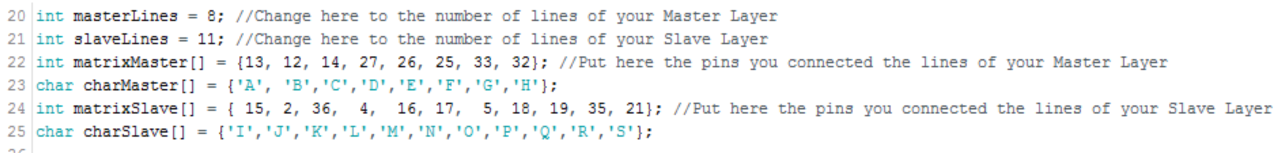
The following is the wiring diagram for a successful setup using an esp32:



Pin G0 was initially used, but proved to be problematic & unreliable. A jumper was used to connect that connector pin (Slave p3) over to SP/G36. Slave pin 10 was routed to G35. Since these pins have no internal pull up, a 10k pull up to 3.3v was tied to each.



The following arrays in the Arduino sketch create a matrix of codes that will be sent via serial when a hit is detected.

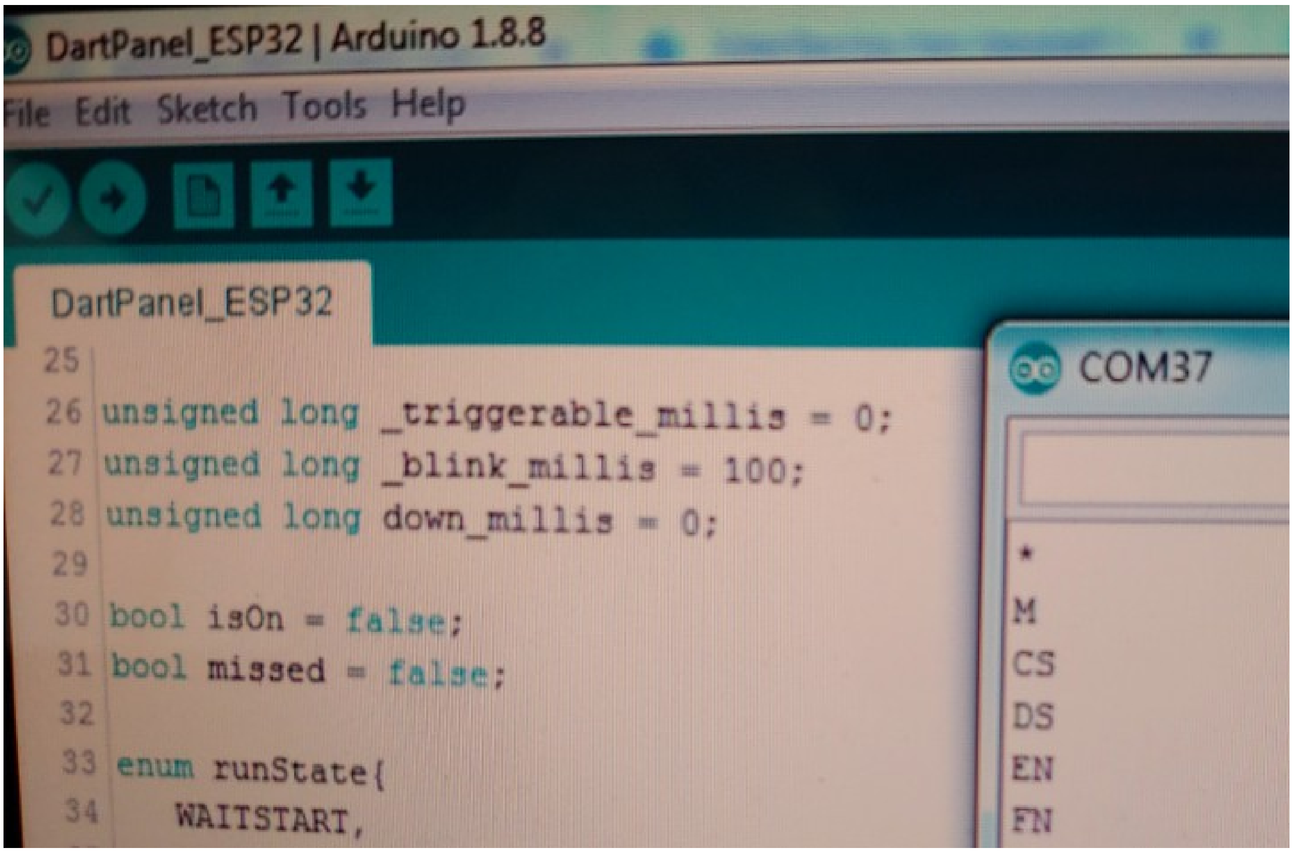


```
int masterLines = 8; //Change here to the number of lines of your Master Layer
int slaveLines = 11; //Change here to the number of lines of your Slave Layer
int matrixMaster[] = {13, 12, 14, 27, 26, 25, 33, 32}; //Put here the pins you connected the lines of your Master Layer
char charMaster[] = {'A', 'B','C','D','E','F','G','H'};
int matrixSlave[] = { 15, 2, 36,  4,  16, 17,  5, 18, 19, 35, 21}; //Put here the pins you connected the lines of your Slave Layer
char charSlave[] = {'I','J','K','L','M','N','O','P','Q','R','S'};
```

For example, when master line 1 is set low and a low is detected on slave line 1, the code AI would be sent from the Arduino to the PC app. This will then be translated by the PC app into the appropriate dart score/hit.

PieMap Configuration

Once the microcontroller is wired and the sketch is uploaded, you can use the Serial Monitor to verify that dart hits are being correctly registered.



```
unsigned long _triggerable_millis = 0;
unsigned long _blink_millis = 100;
unsigned long down_millis = 0;

bool isOn = false;
bool missed = false;

enum runState{
    WAITSTART,
```

```
*
M
CS
DS
EN
FN
```

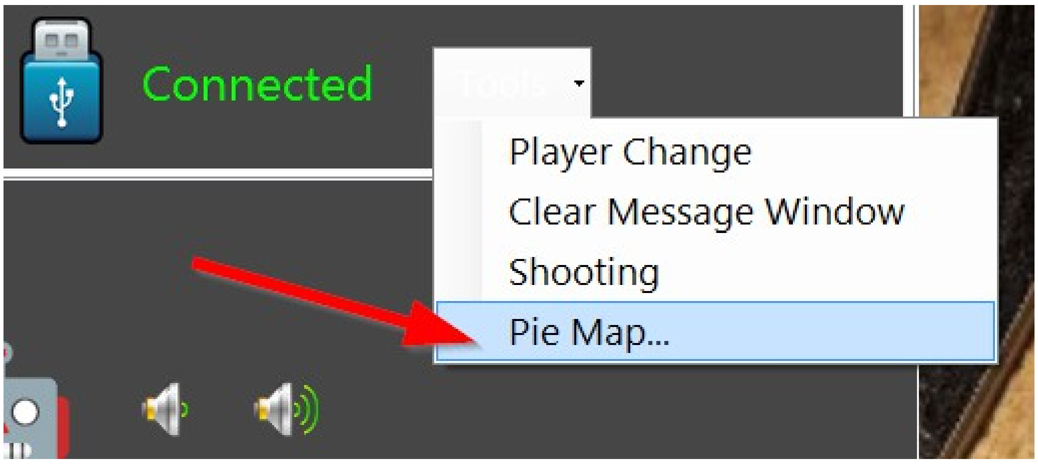
Remember to first enter a C in the top line and hit enter, then enter S and hit enter.

The * in the first line of the Serial Monitor above was the response when the C was entered. The M in the second line is a test of the miss detector. I struck the board on the side in order to make the audio detector sense a noise.

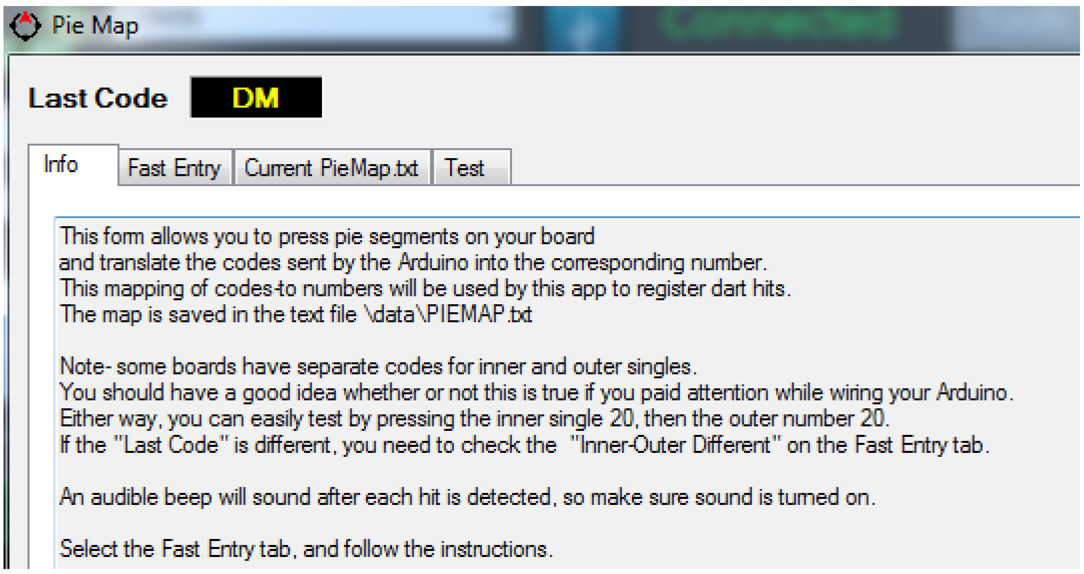
If you see a series of 2 digit codes upon striking pie wedges, you should be good.

You may now proceed to the DartPanel.exe app to set up your dart board configuration.

Open the app with the microcontroller connected. This will populate the Com Port in the settings drop down box. Click the USB icon to connect.



This will connect the PC app to your microcontroller. This step needs to be done every time you open the PC app. From the Tools menu, select "Pie Map...". This will open a dialog where you will be able to configure your board. This only needs to be done once. Follow the instructions on the screens.

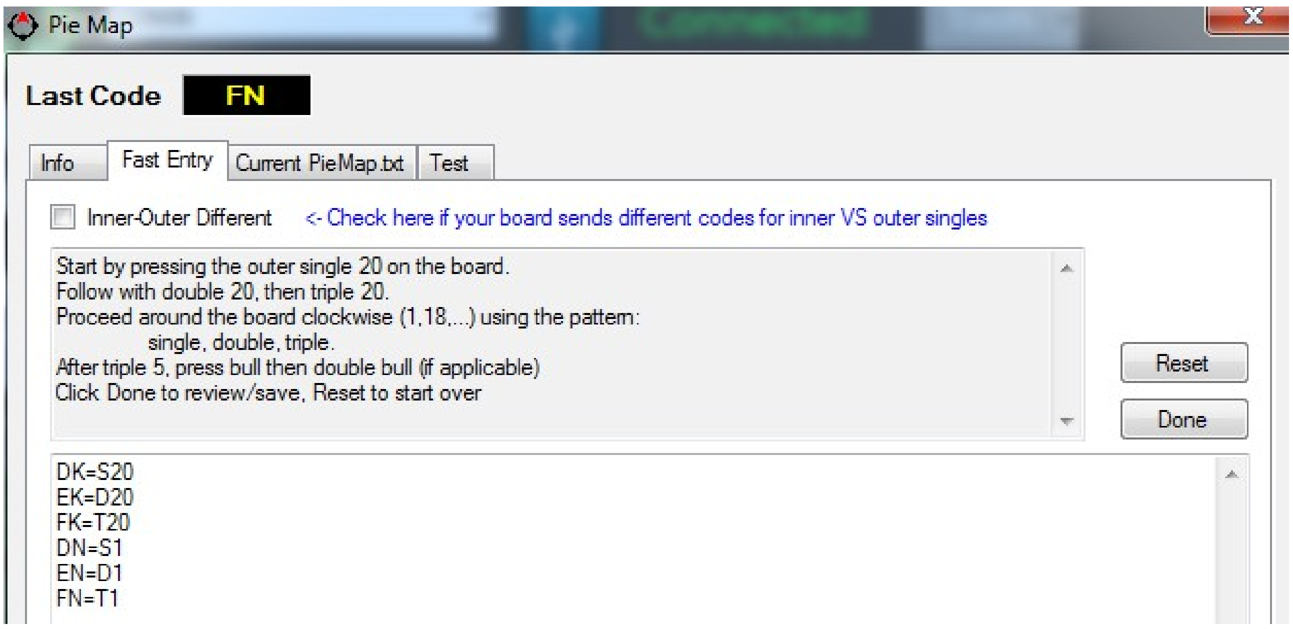


This form allows you to press pie segments on your board
and translate the codes sent by the Arduino into the corresponding number.
This mapping of codes to numbers will be used by this app to register dart hits.
The map is saved in the text file \data\PIEMAP.txt

Note- some boards have separate codes for inner and outer singles.
You should have a good idea whether or not this is true if you paid attention while wiring your Arduino.
Either way, you can easily test by pressing the inner single 20, then the outer number 20.
If the "Last Code" is different, you need to check the "Inner-Outer Different" on the Fast Entry tab.

An audible beep will sound after each hit is detected, so make sure sound is turned on.

Select the Fast Entry tab, and follow the instructions.



Start by pressing the outer single 20 on the board.
Follow with double 20, then triple 20.
Proceed around the board clockwise (1,18,...) using the pattern:
single, double, triple.
After triple 5, press bull then double bull (if applicable)
Click Done to review/save, Reset to start over

```
DK=S20
EK=D20
FK=T20
DN=S1
EN=D1
FN=T1
```

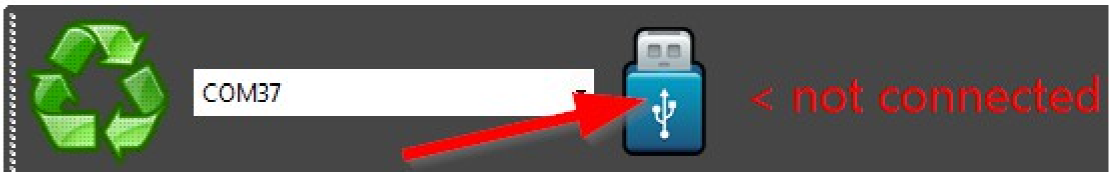
Sample hits being registered. Again, just follow the instructions.

Once you finish and hit "Done", you will be taken to the Current PieMap.txt tab. Review the contents and manually adjust if needed (likely not if you followed the instructions). Then press the "Save" button on that tab. You may now exit this dialog by clicking the X in the upper right.

At this point, you should now be ready to play a game.

Dart Panel Application

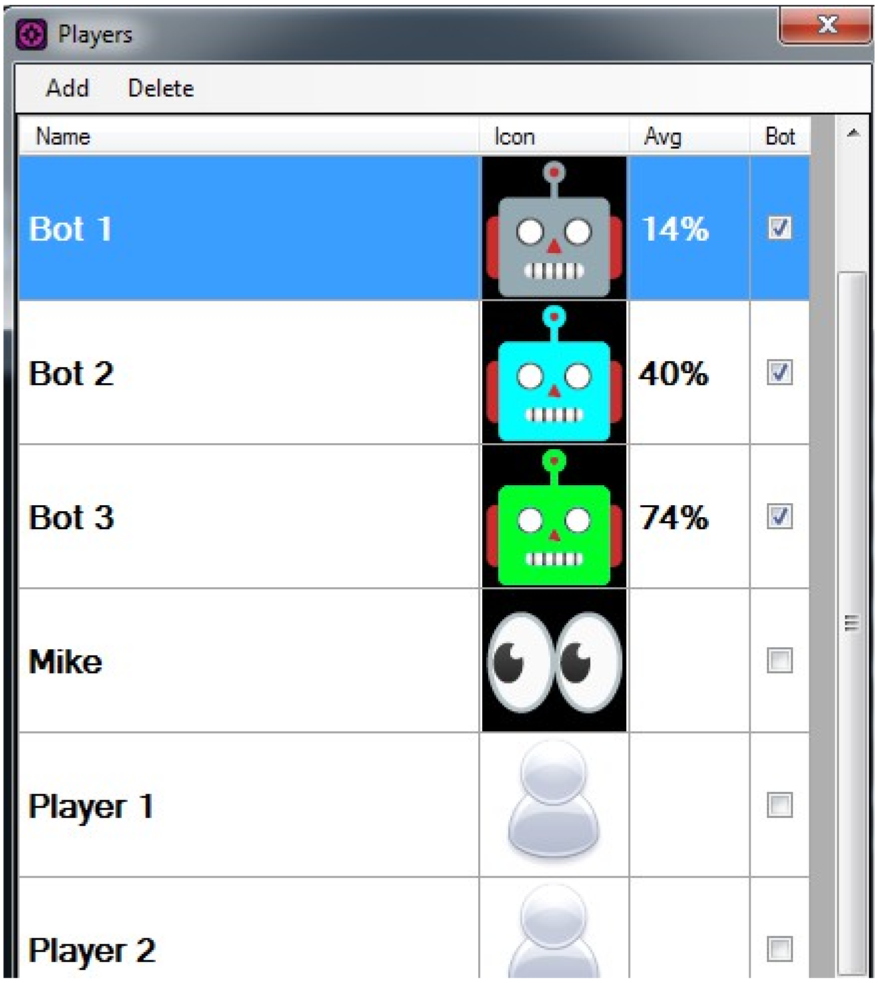
The Dart Panel app always starts at the Settings screen. This is because the first thing you should normally do is connect to your board. Validate the com port, and click the USB icon.



If you plug in your dart board after opening the app, your com port list will be blank. Click the refresh button to scan for serial devices.

Settings Menu Buttons

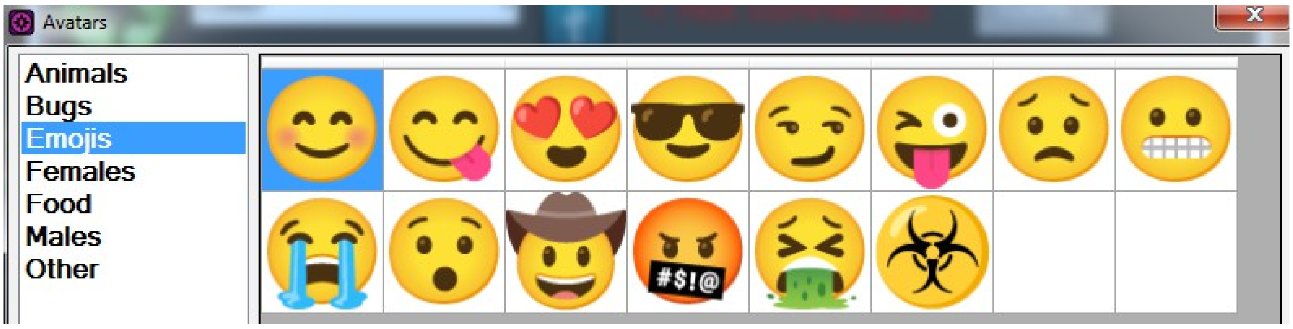
Players- this button takes you to the screen where players may be added, modified, and deleted.



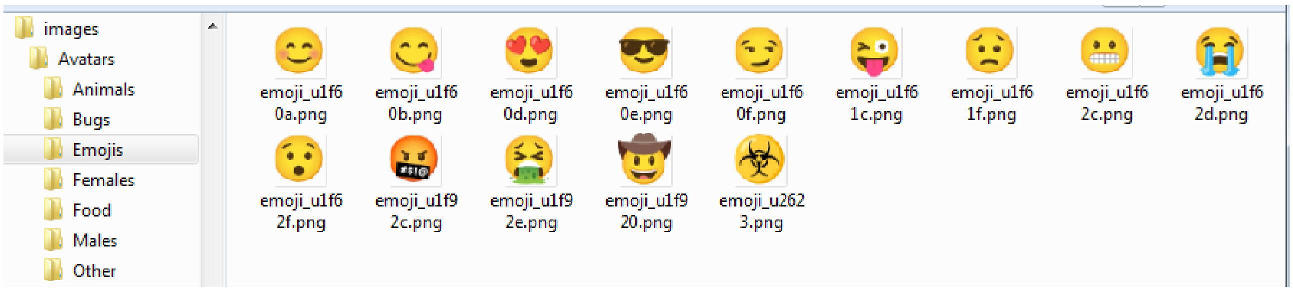
Name	Icon	Avg	Bot
Bot 1		14%	☑
Bot 2		40%	☑
Bot 3		74%	☑
Mike			☐
Player 1			☐
Player 2			☐

The app comes pre-loaded with the following players: Bot 1, Bot 2, Bot 3, Player 1, Player 2, Player 3, and Player 4. These are "System" players, so they may not be deleted.

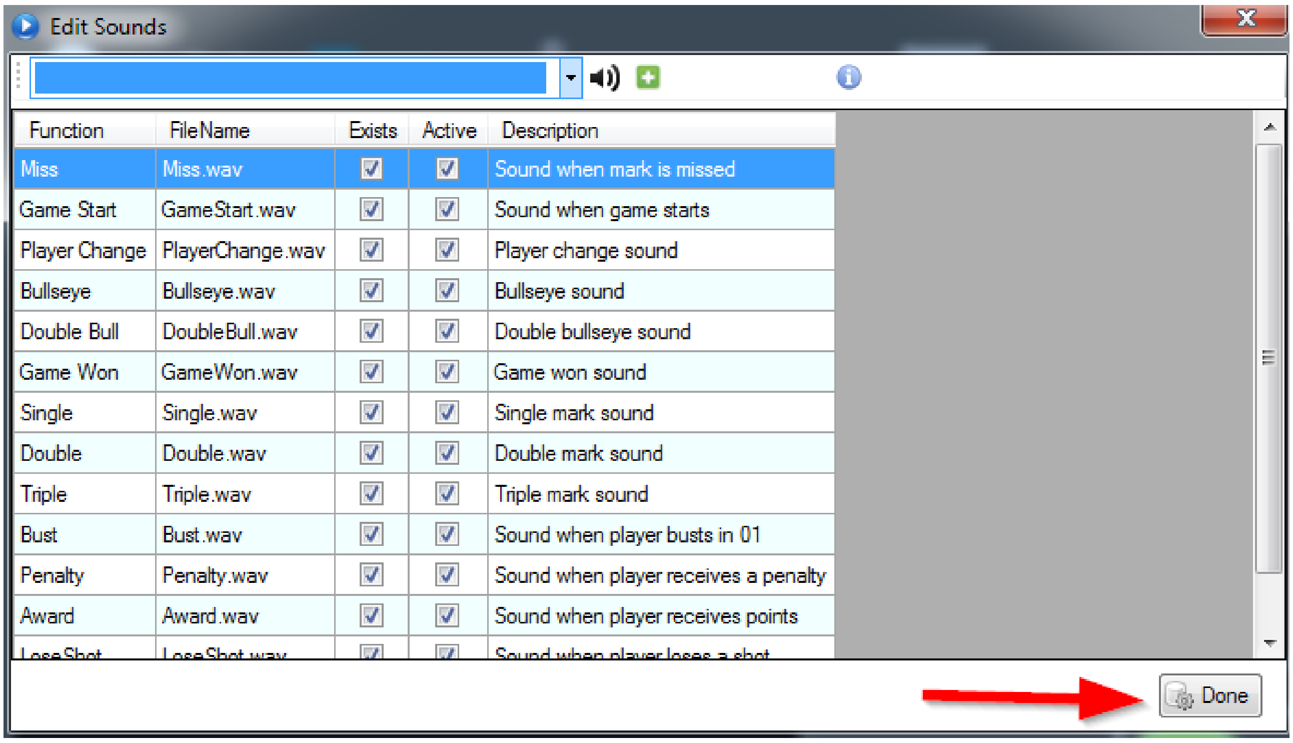
You may change the icon of any player by clicking on its icon. This will take you to a browser where you may select a new icon. To select, double-click.



The screen above reflects the contents of the Avatars subfolders. If you want a new avatar of your own, copy a new .png image into one of the subfolders. It should then be available in the app.



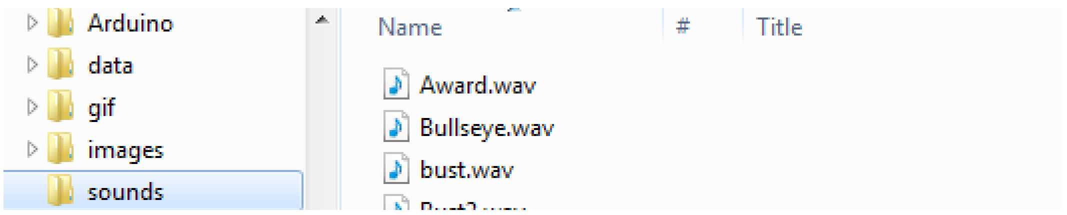
Sounds- This button takes you to the form where sounds are edited.



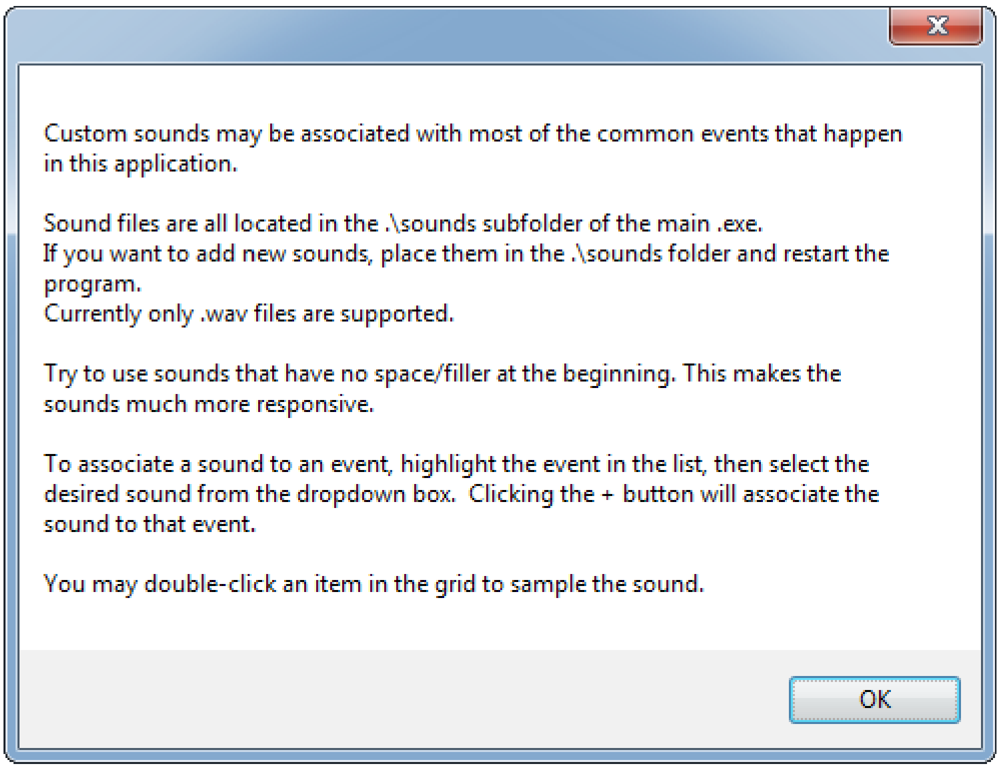
Function	FileName	Exists	Active	Description
Miss	Miss.wav	✓	✓	Sound when mark is missed
Game Start	GameStart.wav	✓	✓	Sound when game starts
Player Change	PlayerChange.wav	✓	✓	Player change sound
Bullseye	Bullseye.wav	✓	✓	Bullseye sound
Double Bull	DoubleBull.wav	✓	✓	Double bullseye sound
Game Won	GameWon.wav	✓	✓	Game won sound
Single	Single.wav	✓	✓	Single mark sound
Double	Double.wav	✓	✓	Double mark sound
Triple	Triple.wav	✓	✓	Triple mark sound
Bust	Bust.wav	✓	✓	Sound when player busts in 01
Penalty	Penalty.wav	✓	✓	Sound when player receives a penalty
Award	Award.wav	✓	✓	Sound when player receives points
LoseShot	LoseShot.wav	✓	✓	Sound when player loses a shot

You may uncheck the "Active" box on any sound to disable that sound.

You may add your own custom sounds to the \sounds subfolder.



From the "i" info button:


Custom sounds may be associated with most of the common events that happen in this application.

Sound files are all located in the .\sounds subfolder of the main .exe.
If you want to add new sounds, place them in the .\sounds folder and restart the program.
Currently only .wav files are supported.

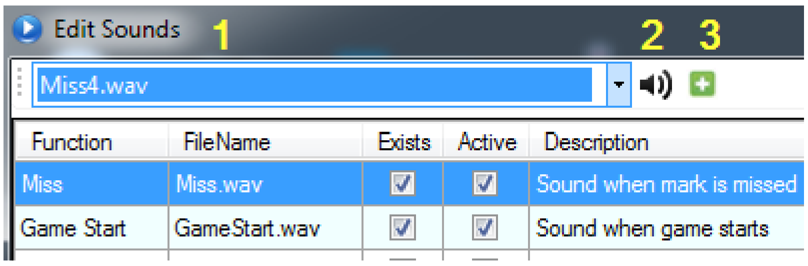
Try to use sounds that have no space/filler at the beginning. This makes the sounds much more responsive.

To associate a sound to an event, highlight the event in the list, then select the desired sound from the dropdown box. Clicking the + button will associate the sound to that event.

You may double-click an item in the grid to sample the sound.

To change a specific sound, select that sound in the list then:

1) Pick a new sound from the list at the top
2) Listen to the sound to make sure it is what you want
3) Click the + button



Function	FileName	Exists	Active	Description
Miss	Miss.wav	✓	✓	Sound when mark is missed
Game Start	GameStart.wav	✓	✓	Sound when game starts

View Videos- This opens the form where animated GIF files may be viewed.

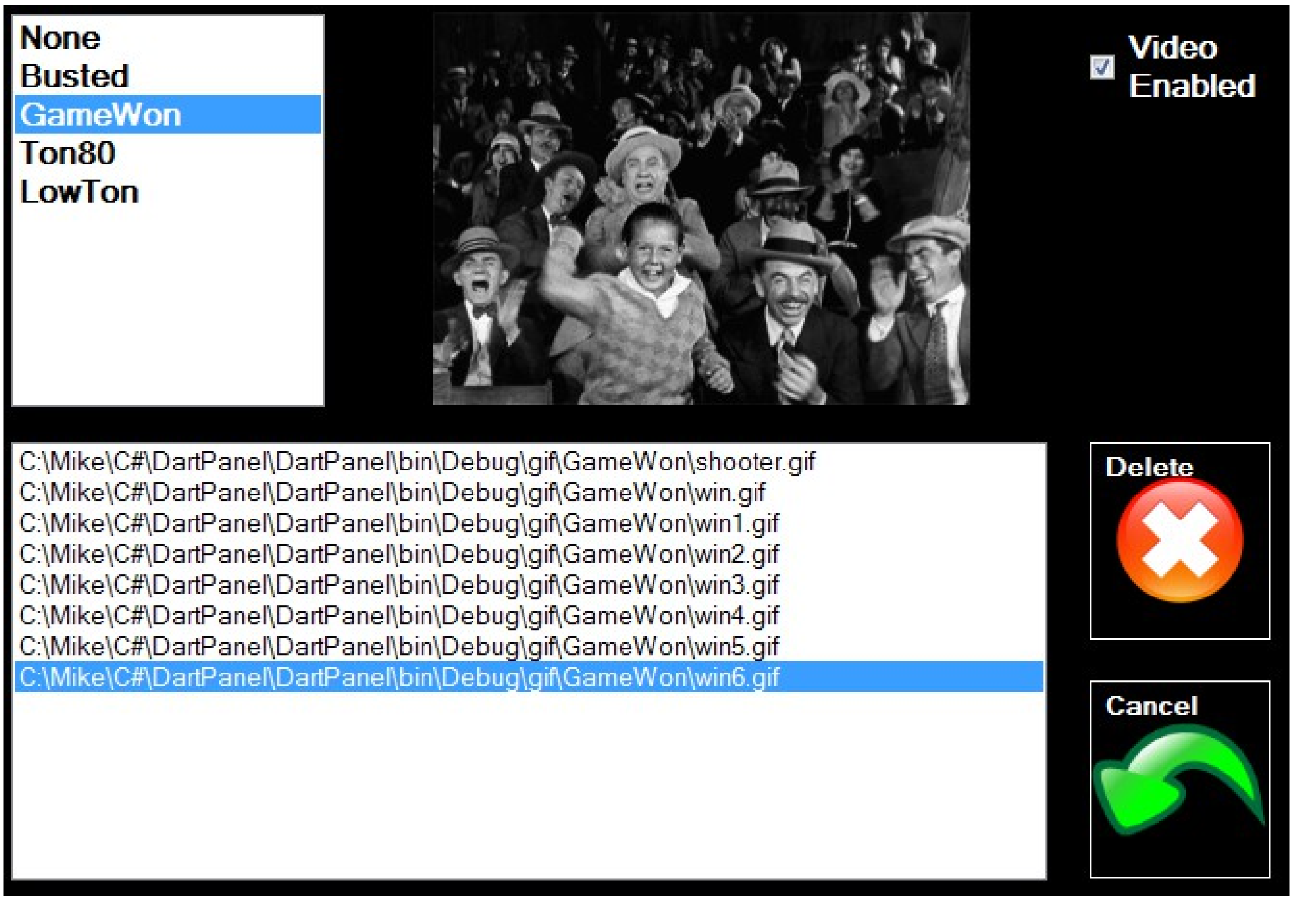
Videos are played when the following events happen:

Busted- If a player busts in an 01 game

GameWon- Video played immediately after a player wins a game

Ton80- Player scores 180 points in an 01 game

LowTon- Player scores >= 100 points in an 01 game



None
Busted
GameWon
Ton80
LowTon

[✓] Video Enabled

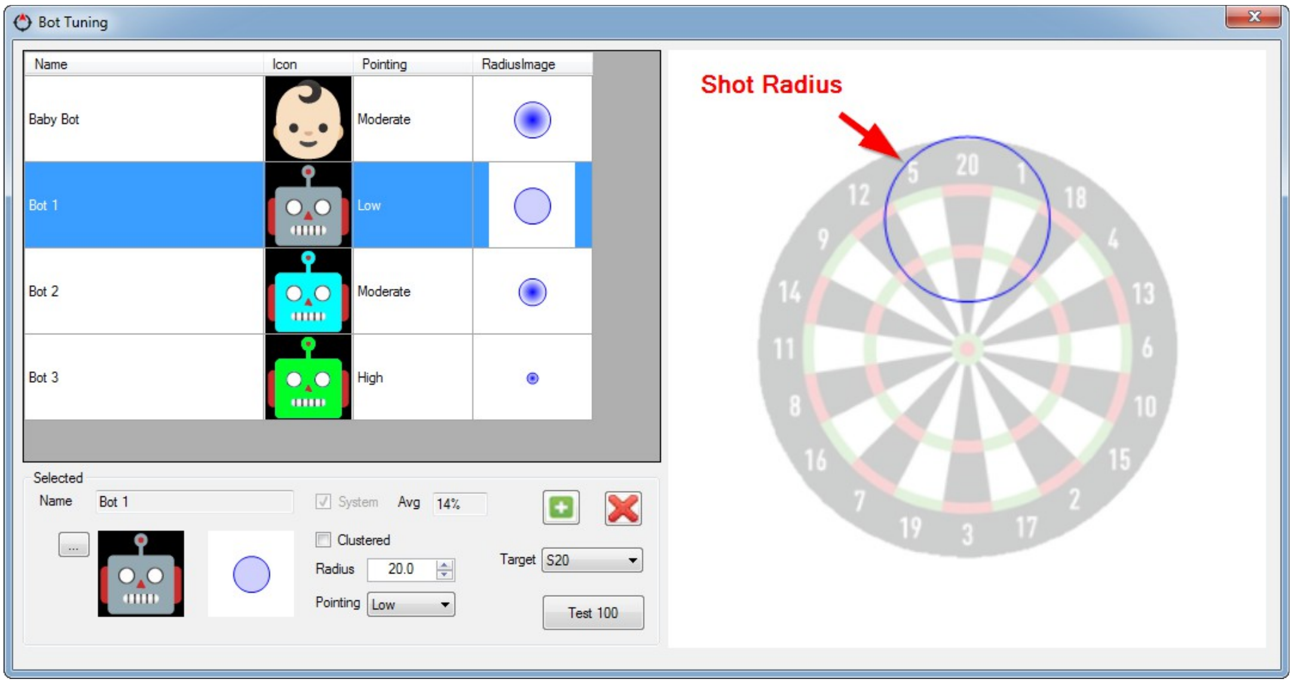
```
C:\Mike\C#\DartPanel\DartPanel\bin\Debug\gif\GameWon\shooter.gif
C:\Mike\C#\DartPanel\DartPanel\bin\Debug\gif\GameWon\win.gif
C:\Mike\C#\DartPanel\DartPanel\bin\Debug\gif\GameWon\win1.gif
C:\Mike\C#\DartPanel\DartPanel\bin\Debug\gif\GameWon\win2.gif
C:\Mike\C#\DartPanel\DartPanel\bin\Debug\gif\GameWon\win3.gif
C:\Mike\C#\DartPanel\DartPanel\bin\Debug\gif\GameWon\win4.gif
C:\Mike\C#\DartPanel\DartPanel\bin\Debug\gif\GameWon\win5.gif
C:\Mike\C#\DartPanel\DartPanel\bin\Debug\gif\GameWon\win6.gif
```

You may add your own GIF files to the \gif subfolders. After adding a new GIF, use this form to view the GIF. This will ensure that the file is not corrupt, etc.

If you want to disable all videos, uncheck the "Video Enabled" checkbox.

If more than 1 file exists in a given folder, a video will be picked at random out of the list.

Bot Training- This is where bots are added, modified, and deleted.



Name	Icon	Pointing	RadiusImage
Baby Bot		Moderate	●
Bot 1		Low	○
Bot 2		Moderate	●
Bot 3		High	●

Selected

Name: Bot 1

System ✓ Avg 14%

Clustered ☐

Radius 20.0

Pointing Low

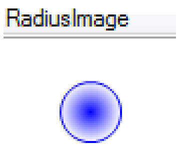
Target S20

To add a new bot, click the + button. Click the X to delete.

How Bots Shoot, and Adjusting Accuracy

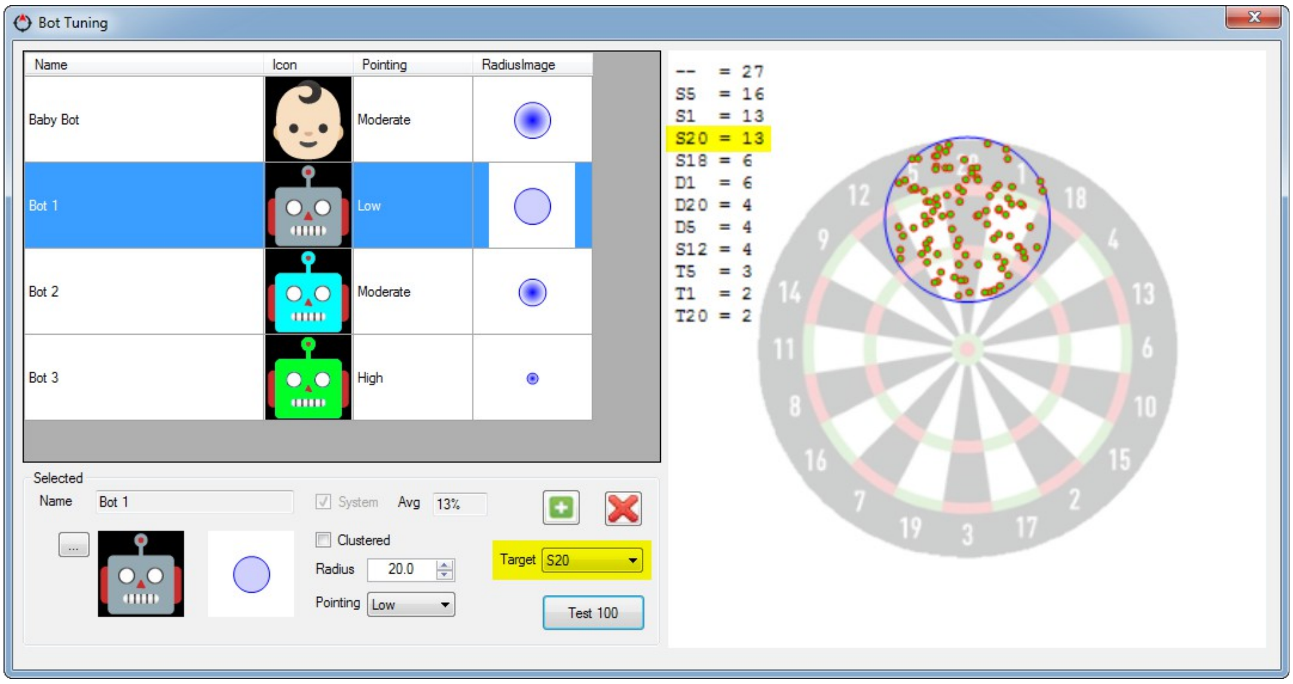
This app uses a novel approach to how bots vary in skill level. Each bot has what is referred to as a "Shot Radius". The smaller the radius, the better the accuracy of the bot. When a bot goes to shoot, the shot they make will be a random point inside their shot radius. One nice thing about this approach is that it creates a more human-like bot performance. When a bot misses the intended number, the misses are typically adjacent numbers, so misses are also more realistic.

The "Clustered" checkbox is used to optionally cluster random shots towards the center of the shot radius. This makes for a more accurate bot. The bots with Clustered checked have a dark center in the sample radius image:



The pointing option of Low, Medium, High is used in Cricket. This helps drive the bot's decision to rack up points VS closing marks.

The "Test 100" button will throw 100 random test darts and list the hit results. This is useful to test the accuracy of a bot's current settings. You may change the number being targeted, default is single 20.


Bot Tuning

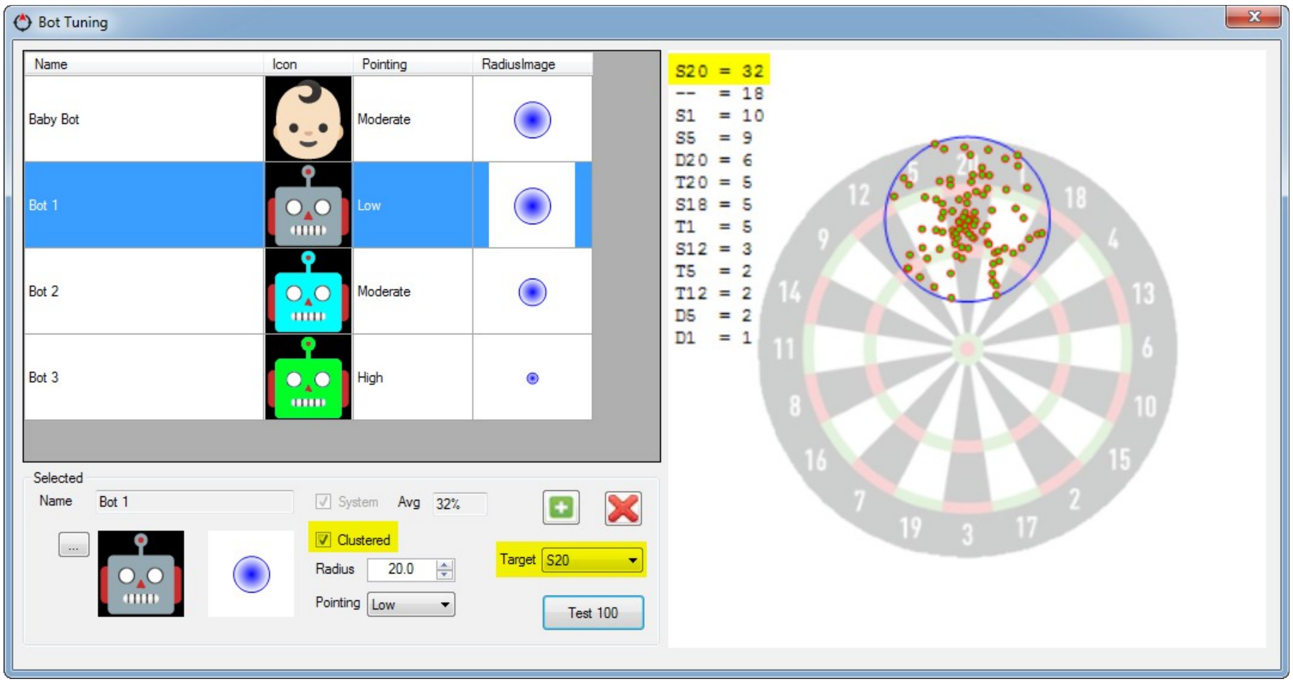
Name	Icon	Pointing	RadiusImage
Baby Bot		Moderate	
Bot 1		Low	
Bot 2		Moderate	
Bot 3		High	

Selected
Name: Bot 1 · System ✓ · Avg: 13%
Clustered ☐ · Radius: 20.0 · Pointing: Low
Target: S20 · Test 100

```
--   = 27
S5   = 16
S1   = 13
S20  = 13
S18  = 6
D1   = 6
D20  = 4
D5   = 4
S12  = 4
T5   = 3
T1   = 2
T20  = 2
```

In the test above, only 13 of 100 test darts found the target of single 20.

With clustering turned on, the hit rate more than doubled. Note the hit pattern is less random:

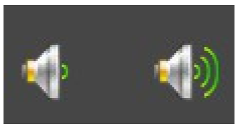

Bot Tuning

Name	Icon	Pointing	RadiusImage
Baby Bot		Moderate	
Bot 1		Low	
Bot 2		Moderate	
Bot 3		High	

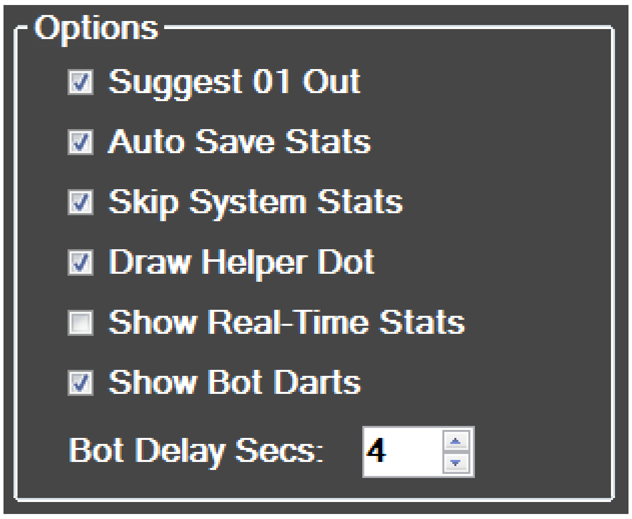
Selected
Name: Bot 1 · System ✓ · Avg: 32%
Clustered ✓ · Radius: 20.0 · Pointing: Low
Target: S20 · Test 100

```
S20  = 32
--   = 18
S1   = 10
S5   = 9
D20  = 6
T20  = 5
S18  = 5
T1   = 5
S12  = 3
T5   = 2
T12  = 2
D5   = 2
D1   = 1
```

Change bot diameters and play with the settings. I like to make Bot 1 a novice, Bot 2 moderately good, and Bot 3 very good. When adding new bots, choose a name that reflects skill level. Lastly, when a Test 100 is performed on the single 20, the hit percentage is saved to the bot profile. That can be used when picking a bot to play against (higher the number, the better they are).



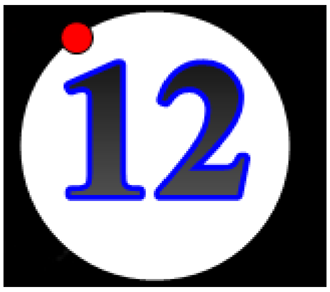
System Volume- These buttons adjust audio volume at the system level.



Options
- ✓ Suggest 01 Out
- ✓ Auto Save Stats
- ✓ Skip System Stats
- ✓ Draw Helper Dot
- ☐ Show Real-Time Stats
- ✓ Show Bot Darts

Bot Delay Secs: 4

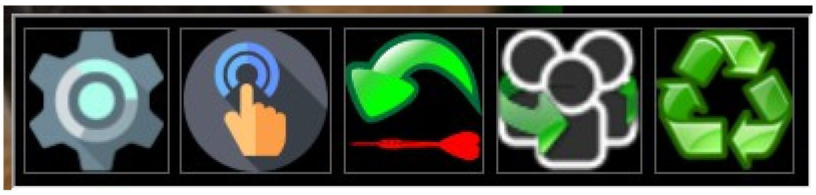
- **Suggest 01 Out**- displays a suggested dart out in 01 games.
- **Auto Save Stats**- Cricket and 01 games collect stats. Having this checked will automatically save these stats in the background when a game is completed.
- **Skip System Stats**- Checking this will make it so that System players (Bot 1, Player 1, etc..) will not have their stats saved.
- **Draw Helper Dot**- This draws a small red dot on the edge of the target number (in Bermuda Triangle, Baseball, and Rando). This dot is oriented to where the target number is on the board. It helps you quickly find numbers that are not commonly shot at, 12 for instance.



- **Show Real-Time Stats**- in Cricket and 01 games, this will show the current shooter's rolling MPR / PPR stats. Not really useful, but available if you want to see it.
- **Show Bot Darts**- When this is checked, and a bot is throwing, a virtual dart board is drawn on the lower-right of the screen. This shows where the bot was aiming, and where their actual dart hit. It was used mostly for audit purposes while writing this app, but also fun to look at. It makes watching the bot throw seem a little more realistic.
- **Bot Delay Secs**- This is the delay (in seconds) between bot throws. 4 or 5 is good.

Main Button Bar

The button bar at the bottom-right of the main screen may be hidden by moving the mouse cursor off-screen to the right. This clears up the screen for playing games.

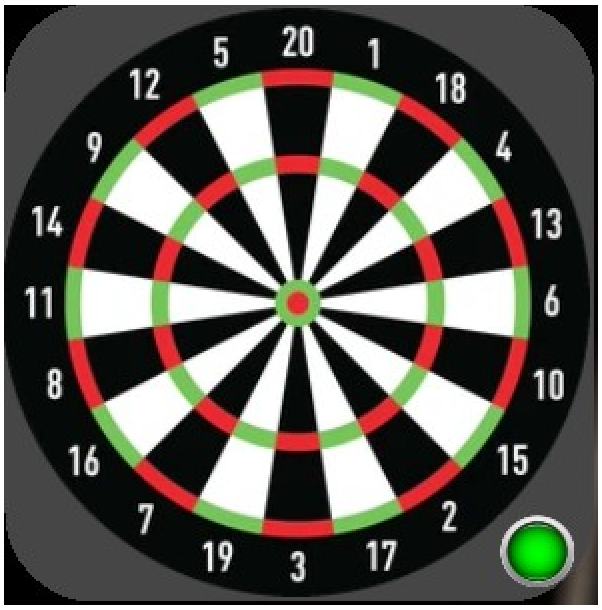


The button functions, in order:

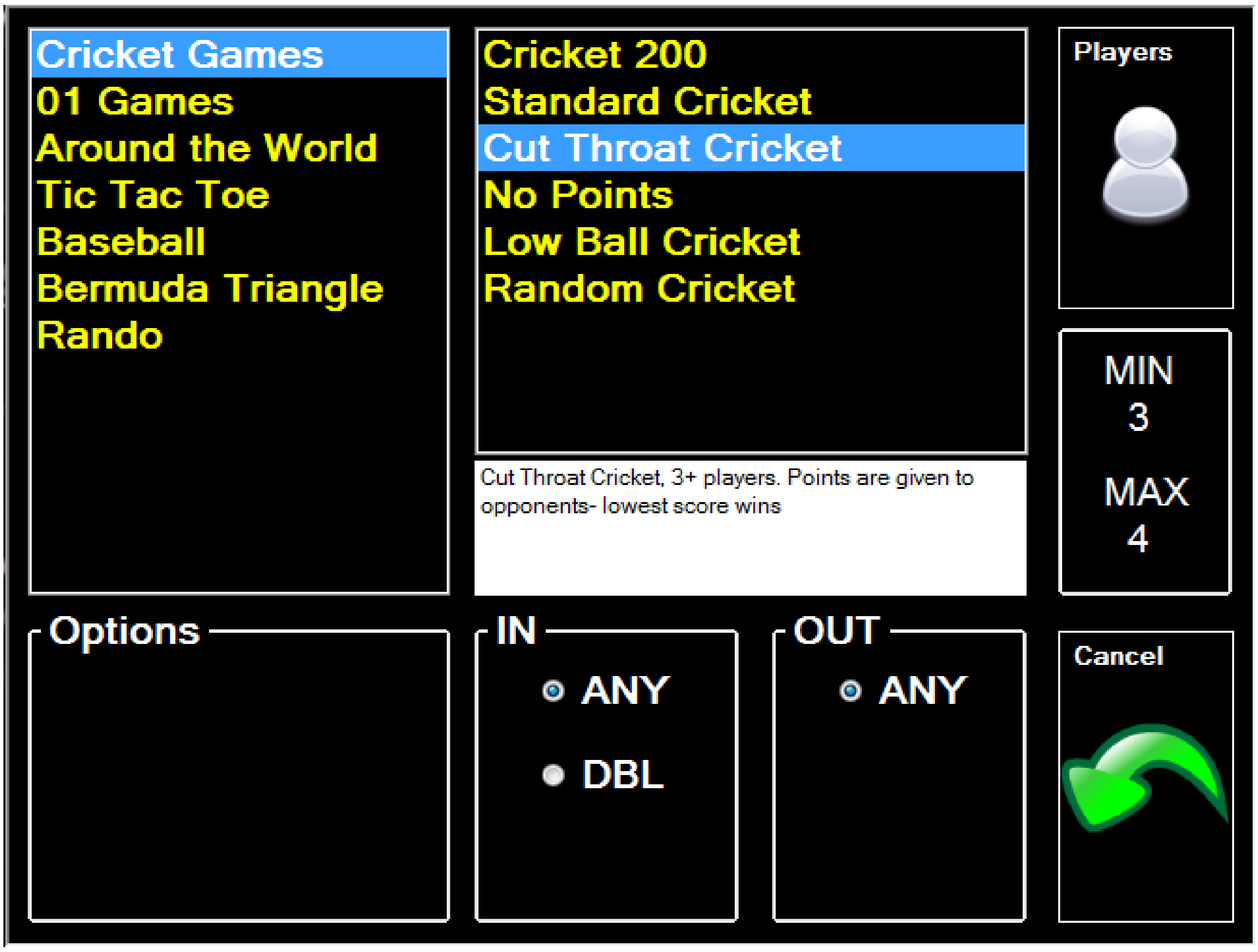
- **Show the settings form**
- **Select game**
- **Undo last dart**
- **Player Change**
- **Reset Game**

Virtual Board

Clicking on the main form will display a virtual dart board. This board may be used to manually register a dart hit, or press player change (the green button). This was mainly used for testing. Clicking on the main form again will hide the board.



Games



Cut Throat Cricket, 3+ players. Points are given to opponents- lowest score wins

The game selection screen is pretty self explanitory. Selecting a game type on the left will give a list of sub-game types to the right. When any sub-game option is selected, a brief description of the game is displayed. Cricket and 01 rules are beyond the scope of this document, but some of the options for novelty games will be described.

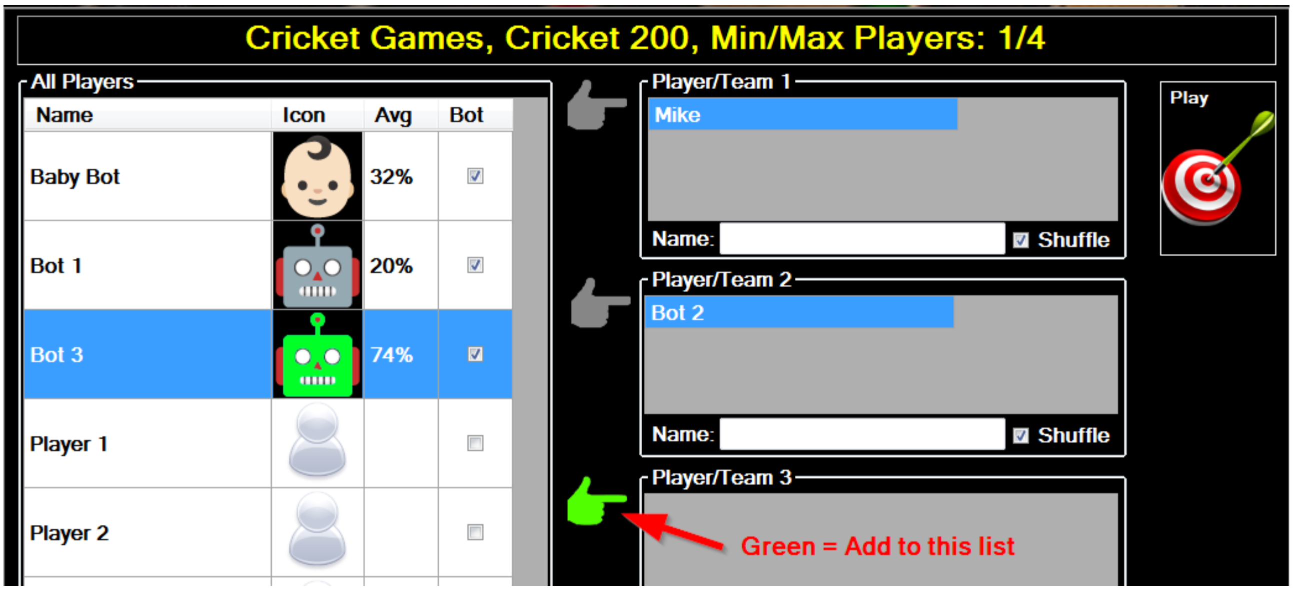
Around the World- The "Keep Going" option is checked by default. When a player hits the target number on the 3rd dart, they get to keep throwing until they miss.

Baseball- The "7th Inning Stretch" option will cut a team's score in half if no runs are scored during the 7th inning.

Rando- This game is similar to "Horse". Each team tries to hit a random number as many times as possible. Team with the most hits for that number wins the point. First to 5 wins. Note, when a target number is displayed, the number is specific. Meaning, if 20 is displayed, this means single 20. If a double or triple is hit, it does not count.

After game options are selected, the next step is to choose players. Click the "Players" button in the upper-right of the screen.

You may choose up to 4 players/teams. Teams may have multiple players.



Name	Icon	Avg	Bot
Baby Bot		32%	✓
Bot 1		20%	✓
Bot 3		74%	✓
Player 1			
Player 2			

Player/Team 1: Mike
Name: ✓ Shuffle

Player/Team 2: Bot 2
Name: ✓ Shuffle

To assign a player to a team, double-click the player on the "All Players" list. This will transfer the player to the list that currently has the green pointer to the left of it. Click the gray pointers to select a particular list.

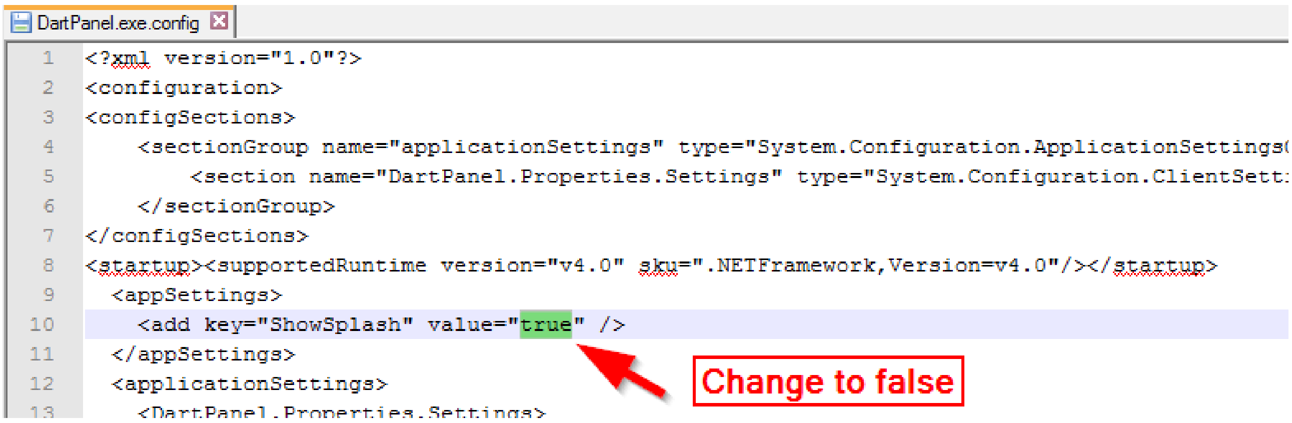
To remove a player from one of the teams on the right, double-click on it. The player will be added back to the "All Players" list on the left.

Click "Play" to start the game.

During game play, after 3 darts are thrown, press the player change button on the dart board before removing darts.

Misc Notes

Note- the initial start up splash screen may be disabled by editing the following file/value:



```
<?xml version="1.0"?>
<configuration>
<configSections>
    <sectionGroup name="applicationSettings" type="System.Configuration.ApplicationSettings(
        <section name="DartPanel.Properties.Settings" type="System.Configuration.ClientSett:
    </sectionGroup>
</configSections>
<startup><supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.0"/></startup>
  <appSettings>
    <add key="ShowSplash" value="true" />
  </appSettings>
  <applicationSettings>
    <DartPanel.Properties.Settings>
```