Bally

Bally Star Trek Pinball
Arduino Game Version ST2025.04
by Dave's Think Tank

Star Trek ST2025.04

Star Trek ST2025.04 re-imagines the original Bally pinball game with new rules, goals, music, light patterns, and sound clips from the TV series and films! Just plug it in, and get ready to experience the game like you've never seen it before!

Note: Requires an Arduino Mega 2560 and a WAV Trigger sound board. See "Getting Started" below.

New Rules

- **Skill Shot:** The next letter you need to finish the word B-A-L-L-Y will be flashing, and is your target for the skill shot. Your reward is 3000 points, in addition to the usual 300 or 3000 points. But that's not all! Hit the 10-point rebound first, and your points are doubled. It's possible to hit the rebound more than once, doubling the score each time! Your points earned are displayed while a Star Trek fanfare plays.
- No flashing letters? If the next letter needed is not A or B or the current saucer value, but the next saucer value, your skill shot is to hit the rebound, advancing the letters, and then land in the saucer for 12,000 points!
- If no letter is flashing, and two or more rebounds would be required to reach the next letter, your skill shot is again to hit the rebound and drop in the saucer for 6,000 points.

- **Klingon Battle:** Pass the second scoring threshold, and you are given one free ball to play a special mini-game against the Klingons! Hit five Klingon ships (five standing targets) for 10,000 points each, plus 50,000 for surviving the battle to the end!
- **Auto-Destruct:** Pass the third scoring threshold, and you're given one ball to save the ship! Alien intruders have activated the ship's auto-destruct sequence, with a sixty second countdown. All four drop targets must be set (for 12,500 points each) in order to turn off the auto-destruct. Your reward for saving the ship is an additional 50,000 points! Watch your time in the credit window.
- **Ball Save:** A particularly bad ball is not held against you! Score less than 5000 points, or play for less than 15 seconds, and Captain Kirk will allow you to "Try again."

New Features

- **Fire Photon Torpedoes!** Complete B-A-L-L-Y, and Captain Kirk will fire photon torpedoes from the Enterprise to celebrate the win!
- **Fire All Phasers!** The bonus is collected at the end of each ball, or when you return to the shooter lane through the warp speed lane. The Enterprise will then fire phasers up to three times, once for each multiple earned.
- **Enterprise Computer:** The Enterprise computer is used to calculate the match score at the end of the game, complete with sound, voice, and flashing lights!
- **New Sounds:** New sound effects and voices make the game more fun and exciting. Music from the original show or bridge sounds play in the background.
- **High High Scores:** If your score exceeded a million, the old game would start your score over at zero. Now high scores over a million are maintained, and displayed in the 6-digit displays(!)

Familiar Rules from the Original Game

- Making B-A-L-L-Y scores and advances lit values.
- All drop targets down scores 5000 points.
  - 1st time lites 2X.
  - 2nd time lites 3X bonus.
  - 3rd time and each additional time scores special.
- Top rollover buttons and lit flipper feed lanes score and advance hyperspace values.
- Outlanes special – Lites when B-A-L-L-Y special is lit.
- Ball thru outlane when lit scores special.
- Ball thru Warp Speed Lane (into shooter lane) scores bonus.
- Maximum 1 extra ball per ball in play.
- Tilt penalty – ball in play.
- Three threshold scores can be set. The first earns points, an extra ball, or a free game. The second and third activate the Klingon Battle and Auto-Destruct mini-games.
- 3 or 5 balls per game.
- Above rules can be adjusted with DIP switches and self-test settings.

Getting Started

Purchase Arduino with WAV Trigger: https://pinside.com/pinball/shops/shop/1304-roygbev-pinball/13139-concord2022-with-new-sounds-complete-kit
Purchase Arduino IDE: https://www.arduino.cc/en/software (if software needs to be installed)

Install the Arduino and WAV Trigger boards as per instructions included with device. These instructions are also included at the end of this manual. If your Arduino came with the ST2025.04 software included, you're done! (Currently it is not sold this way, and you will have to install the software as described below.) Note: The Arduino can also be programmed before installing in the pinball, if you prefer.

Create a folder named ST2025p04 on your computer. Download the software from https://github.com/DavesThinkTank/Bally-Star-Trek to this folder.

The sound files can be found at:
https://drive.google.com/drive/folders/175rKGxsXPs678i7x1qTePkK48J6tJLC7?usp=sharing

Transfer the included sound files to the micro-SD card for your WAV Trigger board (into the root directory).

TURN OFF your pinball machine (if programming the Arduino after installation)! Make sure "Switch" connectors on the Arduino are connected with the included jumper.

Plug the Arduino into a USB port on your laptop computer with an appropriate cable. You need a cable with a USB micro connector on one end, and a USB A or C on the other end, whichever your computer needs. Ignore the LED lights on your pinball's circuit boards. It's normal for the Arduino to power some LEDs.

Open the file ST2025p04.ino (in the ST2025p04 folder) with the IDE by double-clicking on it.

Click on the white box at the top left of the IDE. Click on "Select other board and port". Select "Arduino Mega or Mega 2560", and the port your Arduino board is plugged into.

Click on the Verify checkmark (beside white box) to make sure software compiles properly. Click on the Upload arrow (also beside white box) to compile the software and upload to the Arduino.

Unplug cable from computer.

Unplug the old speaker from the sound board in the backbox!

The game should now run on your pinball. When you turn on the pinball, the first thing you should see is the version number; 2025 in the player 1 display, and 04 in the credit window. This will last for about four seconds. If you don't see this, you haven't got the software running yet! Also, if it's not the latest version number, go back to GitHub and find the latest release.

RUN THROUGH ALL THE SELF-TESTS AND GAME SETTINGS BEFORE PLAYING YOUR FIRST GAME! Make sure everything is set the way you want it. See the self-test section below.

Read the included manual, and the included readme.md file, and watch the following YouTube video for more assistance: https://youtu.be/hn4zS7xABDA?si=LBjIKUdJgUR4hjqQ

Arduino Self-Tests

The Arduino self-test is similar to the regular Bally self-test. You begin by pressing the red self-test button inside the coin door.

There are getting to be a LOT of tests, audit settings, and game settings. You can now use the slam switch (on the inside of the coin door) to end self-test at any point (other than during the switch test), and return to attract mode.

At any point within the Self-Test or Attract Mode (again, other than during the switch test), you can enter the new Kids' Mode by pressing the Game Button and Coin Slot 3 switch at the same time (hold until you hear the sound). Kids' Mode changes all the DIP switch and self-test game settings to easy levels. Put the game back to regular game mode by pressing the two buttons at the same time again, or by simply turning the machine off. Audio cues let you know which mode you have entered. See the section titled "Kids' Mode" below for more information.

Light Test

Ball in Play: Test #01
Display #1: Light number, or 99 for all lights
Game Button: Cycle through switched Illumination lights. Hold to cycle continuously.

The first test will repeatedly flash all the switched illumination lights on the playfield and in the backbox. This is similar to the regular Bally light test, except the Arduino allows you to now press the game button on the front of the coin door. When you do so, all the lights will stop flashing except one. By continuously pressing the game button, the pinball will cycle through all the lights, displaying each, one at a time. Display #1 shows the corresponding light number. A table of all the lights and the sequence in which they appear is included below.

Display Test

Display #1-5: All digits cycle through numbers 0-9
Game Button: Cycle through individual digits. Hold to cycle continuously.

Pressing the self-test button again will then take you to the display test. Again, this is similar to the Bally display test in that it cycles all digits in all five displays through the numbers from 0 to 9 repeatedly. It cycles quite a bit faster than the Bally test though, making this a much less tedious review!

And again, the Arduino extends this test with use of the game button. When you press the game button, all displays will go blank except for the first digit on the first display, which will continue to cycle. Pressing it again moves this to the second digit. Pressing it again moves to the third, and so on, going through each digit of each display individually. After the final digit, pressing the game button will set all displays running through the numbers again. Holding the game button down will cycle quickly through each individual digit.

Solenoid Test
Ball in Play: Test #03
Credit Display: Switches firing due to solenoid activity (if any)
Display #1: Solenoid number
Display #4: Time in milliseconds between solenoid firing and switch activating
Game Button: Fire current solenoid repeatedly. Press again to continue cycling.
Coin 3 Switch: Stop solenoids from firing. Clear switch display (credit window).

Pressing the self-test button again takes you to the solenoid test. This runs through all the solenoids, just like the regular Bally test (except in a different order). See the table below for a list of solenoids, and the order used. Note, the Coin Door Lockout and the K1 Relay Flipper Enable are not included in these tests.

New Features:

Pressing the game button at any point will cause the current solenoid to continue firing repeatedly, so you no longer have to cycle through all of the solenoids to see the one you are interested in. Press again to continue cycling. Press the coin 3 switch to turn firing of solenoids off and back on. This allows you to make adjustments to a solenoid while remaining in test mode!

Keep an eye on the credit window during this test. If vibration from a solenoid causes a switch to misfire, the switch number will be displayed here. The time between the solenoid firing and the switch activation is displayed in Display #4 (in milliseconds). See the section “User Programmable Changes” to see how this information can be used to fix this issue.

The flippers are enabled throughout the solenoid test. This is therefore a good place to work on flipper issues.

Stuck Switch Test
Ball in Play: Test #04
Credit Display: The number of switches currently closed
Display #1-4: The lowest four stuck switch numbers
Game Button: Double-click to reset the drop targets

Pressing the self-test again takes you to the switch test. Switches that are stuck on will be identified by number in the displays, like the original test. However, the Arduino allows up to four stuck switches to be identified on four displays. The original Bally test displayed only the lowest-numbered stuck switch, making testing of multiple stuck switches and switch-matrix issues difficult. The number of closed switches is also displayed in the Credit display, for cases where more than four switches are closed at once. The same numbering system is used as the original, as in the table below.

Double-clicking the game button will reset all drop targets during the switch test. This allows you to easily test and work with drop target switches, and then deactivate them again.

Note: In order to allow testing of the slam switch, game button, and coin slot 3 switch, the special functions assigned to these buttons during self-tests do not work during the switch test.

Detecting Switch Matrix Issues

The Stuck Switch test can also be used to locate switch matrix issues. The 40 switches of a pinball are wired together in an 8x5 grid. Diodes on each switch make sure one switch closing cannot affect any other switch, but a bad diode can cause problems. If a closed switch has a bad diode, and another switch in the same row is closed, and another in the same column is closed, then a fourth switch at the opposite intersection of the row and column will also register as closed.

Testing for switch matrix issues:

1. Fix all stuck switches. Make sure all switches are open. All four displays should be blank.
2. Test all switches individually. Make sure you know where they all are. Note that the switch matrix diagram in your schematics may be inaccurate. Note any errors.
3. Start with switch 0. Close the switch, and hold it closed.
4. Choose any other switch in the same row (refer to the switch matrix chart below). Close the switch, and hold it closed.
5. Choose any other switch in the same column. Close the switch, and hold it closed.
6. Three displays should show the three switches you are holding closed. If a fourth display indicates another switch, then switch zero has a bad diode and is causing a switch matrix error. (Note, the coin slot switches do not have diodes and should register as causing a switch matrix error.)
7. If multiple switches are assigned the same switch number, be sure to test them all by opening the one you are holding, and closing the next one.
8. Open all the switches. Proceed to test switch 1, then 2, and so on to switch 39.

Switch Bounce (Double-Hit) Test

Ball in Play: Test #05
Display #1: Most recent switch hit
Display #2: The time between hits in milliseconds
Game Button: Double-click to reset all drop targets

Pressing the self-test button again takes you to the switch bounce test. Switches on your pinball machine may develop a “bounce”, where hitting them registers two or more hits. If you suspect this may be happening with a switch on your machine, this test can help you to identify the issue.

To determine whether a switch is bouncing, activate the suspected switch with a pinball. If it registers only once, the switch number will appear in the Player 1 display, and all other displays will be blank. If it registers two or more times, the time between hits will appear in the Player 2 display (measured in milliseconds). See the section “User Programmable Changes” to see how this information can be used to fix this issue.

Double-clicking the game button will reset all drop targets during the switch test. This allows you to easily test and work with drop target switches, and then deactivate them again.

Sound Test

Ball in Play: Test #06
Display #1: Sound number
Game Button: Play same sound repeatedly. Press again to continue cycling. Press within ½ second of display change to skip current sound. Hold to skip many sounds quickly.

Pressing self-test again takes you to the sound test. The original Bally test simply played a single sound. The Arduino cycles through all the sounds. Pressing the game button plays the current sound repeatedly. Pressing it again will continue cycling sounds.

Display #1 will indicate the sound number to be played. If the game button is pressed within one half second of the display changing, the current sound will be skipped. Holding the button will increase speed, skipping sounds (very useful for the long, empty stretches!). See the table below for a list of sounds.

Be aware, each sound will take five seconds before proceeding to the next. This is a bit slow, but it generally prevents the sounds from running over each other.

Testing and reviewing the sounds can be challenging. If a sound is making it difficult to test subsequent sounds, try the “skip” feature.

DIP Switch Test

Ball in Play: Test #07
Display #1 - 4: DIP switch values (1 = ON, 0 = OFF), first six digits of 4 DIP banks
Credits: DIP switch values for final two digits of the current DIP switch bank
Game Button: Move to next DIP switch. Hold to cycle through switches quickly.
Coin 3 Switch: Change setting of current DIP switch

Pressing self-test again takes you to the DIP switch test. This completely new test shows you the setting of all 32 DIP switches, and allows you to change them until the pinball is turned off. Turning the machine off and on again restores the DIP switches to the settings on the MPU board.

All 32 DIP switches are shown in the 26 (!) display digits as either 1 (ON) or 0 (OFF). Since the displays are only six digits, the first six switches of each bank of eight are shown in the four displays. The 7th and 8th digits of each bank are shown in the Credit window.

The current switch is identified by a flashing number, and the two switches in the Credit display belong to this bank. By pressing the game button, you can scroll through switches 1 to 32. The values in the Credit window will update to show the last two switches of the current bank, as the current bank changes. Stop on a switch and you can use the coin 3 switch to change its setting temporarily. This new setting will continue to be used until the pinball is turned off, or it is reset using this screen again.

This can be useful to detect defective DIP switches, test different DIP switch settings, or set up an easy game temporarily for kids or guests.

Score Levels (Award Levels 01, 02, and 03, High Score, and Personal Goal)

Credit Display: Threshold level #01, 02, 03; 04 for high score; 05 for personal goal
Display #1: Score Amount
Game Button: Increase by 1000. Hold to speed-up the process.
Double Click: Set value to zero

Pressing the self-test button again takes you to the first score level. Reach this level in the game and you can win points, an extra ball, or a free game (see DIP switches 14 and 15 below). Reaching the second score level activates the Klingon Battle mini-game. The third score level activates the Auto-Destruct mini-game. Next is the high score, and then "Personal Goal", a new feature that provides different ending quotes for good or bad games. Set Personal Goal to what you consider a good game.

Press the game button to increase these values by 1000. Hold the button in to increase the value continuously. This is very similar to the original Bally, except by holding the button the score will begin to increase very quickly, a definite improvement! Release it to stop, and press again to start off slowly. Double-pressing the button resets the value to zero. A threshold level of zero means no award for this level or higher.

Score Levels:
01: Threshold Level 1 (50,000 points, extra ball, or free game. See DIP switches 14 and 15 below.)
02: Threshold Level 2 (Klingon Battle mini-game)
03: Threshold Level 3 (Auto-Destruct mini-game)
04: High Score
05: Personal Goal

Accounting Info
Credit Display: Level #06 through 12
Display #1: Value of accounting item
Game Button: Increase value by 1. Hold to increase repeatedly.
Double Click: Set value to zero

Tests 06 through 12 cover number of credits, total number of games played, total number of free games won, number of times high score beat, and number of coins collected in chutes 2, 1, and 3. Hold the game button to increase, or double-click to set these to zero.

Accounting Items:
06: Credits
07: Total plays
08: Total replays
09: Number of times high score beat
10: Chute 2 coins
11: Chute 1 coins
12: Chute 3 coins

Game Settings

Tests 13 through 16 change certain game settings, as outlined below.

Credit Display: Game Setting Number (13 through 16)
Display #1, 2, and 3: Values of settings
Game Button: Increase value
Returns to minimum value after reaching maximum
Hold to increase repeatedly
Double Click: Set value to minimum (or reverse direction for #16)
Coin Slot #3 Switch: Move to next value (on screens with more than one value)

13: Background Sound On / Off
00: Background music or sounds will be silenced
01: Background will play

14: Free Play
00: No free play. Coins must be inserted to play game.
01: Game can be started by pressing the game button, without inserting coins.

If Free Play is selected, the amount in Credits (accounting item 6 above) will determine whether or not the credit light on the apron is lit. Accounting item 6 will then be the only way to turn this light on or off. This could be important, if you have a custom apron which incorporates the credit light into its design!

15: Ball Save Settings

Credit Display: Level #15
Display #1: Number of ball saves per game
Display #2: Maximum score to invoke ball save
Display #3: Maximum play time to invoke ball save
Game Button: Increase value. Hold to increase repeatedly. Double-click to reset to zero.
Coin 3 Switch: Move to next displayed value

“Ball Save” is a feature that will let you play a ball over again, under certain circumstances. This screen allows you to set the number of ball saves allowed per game (maximum 5), the maximum score at which ball save will be invoked (maximum 25000), and the maximum play time at which ball save will be invoked (maximum 25 seconds).

If you have a particularly bad ball, your score will be reset to the beginning of the previous ball, “Same Player Shoots Again” lights will come on, and Captain Kirk will offer you to “Try Again”.

If you do not want ball save, set the number per game to zero, but leave the other values non-zero! Setting all three values to zero will cause the program to think you have not set any values yet, and it will reset them to the default values.

16: Relative Volume Settings (Background Music, Sound Effects, and Voices)

Credit Display: Level #16
Display #1: Volume adjustment for background sounds (0 – 100)
Display #2: Volume adjustment for sound effects (0 – 100)
Display #3: Volume adjustment for voices (0 – 100)
Game Button: Increase or decrease value. Hold to increase / decrease repeatedly.
Double-Click: Switch game button between increasing or decreasing values.
Coin 3 Switch: Move to next displayed value

This test allows you to adjust the relative volumes of the background music (display 1), sound effects* (display 2), and voices (display 3). In general, you want sound effects to be quieter, voices to be louder, and background music to be somewhere in the middle. The volume levels have already been set in this way, and to leave them alone, simply set all three values to 50. Values above 50 will raise the volume for sounds of that type, and values below 50 will lower those volumes.

Please note this screen is used to set the *relative* volume for the three sound types! It therefore makes little sense to raise all the values above 50, or to lower them all below 50. To raise or lower all three, you are far better off using the volume control on your speaker!

Also keep in mind, raising the volume too high on one or more sound types could cause a problem called "clipping", where the highs and lows of the sound are lost or distorted. This can be very evident if you've raised two or all three types too high, and they are all played at the same time and added together. So don't do this.

A piece of background music will play throughout this test, and you can adjust its volume up or down to hear the effects. When adjusting the sound effect or voice setting, a sound effect or voice will play repeatedly over the background music, allowing you to hear the relative effects.

Note that, at first, pressing the game button will raise the volume, and holding it will raise the volume repeatedly. To lower the volume, double-click the game button. Now pressing or holding the game button lowers the volume. Double-click a second time to increase the value again.

* Some sound effects are set to loud, and will be unaffected by the sound effect setting. This includes the pop bumper explosions and the slingshot phasers. They are set to be slightly louder than the voice setting.

Self-Test Information Tables

The following tables can be used, together with the self-test feature, to investigate the functioning of your pinball. These tables will assist you in determining the game feature being indicated by the values displayed during the tests.

List of Arduino Self-Tests

Ball in Play Display	Credit Display	Test / Setting
1		Lights
2		Displays
3		Solenoids
4		Stuck Switches
5		Switch Bounce
6		Sounds
7		DIP Switches
	1	Score Threshold Level 1
	2	Score Threshold Level 2
	3	Score Threshold Level 3
	4	High Score
	5	Personal Goal
	6	Credits
	7	Total Plays
	8	Total Replays
	9	High Score Beat
	10	Coins in Chute 2
	11	Coins in Chute 1
	12	Coins in Chute 3
	13	Background Sound Off or On (0 or 1)
	14	Free Play Off or On (0 or 1)
	15	Ball Save Options
	16	Relative Volume for Background Music, Voices, and Special Effects

Lights

No.	Light	No.	Light
0	Bonus 1	30	Right Flipper Lane
1	Bonus 2	31	Left Flipper Lane
2	Bonus 3	32	B Lane
3	Bonus 4	33	A Lane
4	Bonus 5	34	Klingon L #1
5	Bonus 6	35	Klingon L #2
6	Bonus 7	36	Klingon Y
7	Bonus 8	37	Right Out Lane
8	Bonus 9	38	Left Out Lane
9	Bonus 10	39	Center Y
10	Bonus 20	40	Same Player Shoots Again (Backbox)
11	Not Used	41	Match
12	Not Used	42	Same Player Shoots Again (Playfield)
13	Planet Special + Center 3X	43	Credit Indicator (Apron)
14	Planet 3X + Center 2X	44	Center 10
15	Planet 2X	45	Center 25
16	Hyperspace 2	46	Center 50
17	Hyperspace 4	47	Center Special
18	Hyperspace 6	48	Ball In Play
19	Hyperspace 8	49	High Game
20	Hyperspace 10	50	Game Over
21	Hyperspace Special	51	Tilt
22	Planetary Bypass Inside Lane	52	Center B
23	Planetary Bypass Outside Lane	53	Center A
24	Saucer B	54	Center L #1
25	Saucer A	55	Center L #2
26	Saucer L #1	56	1 Up
27	Saucer L #2	57	2 Up
28	Saucer Y	58	3 Up
29	300 or 3000	59	4 Up

Solenoids

No.	Solenoid
0	Not Used
1	Not Used
2	Not Used
3	Not Used
4	Not Used
5	Knocker
6	Outhole Kicker
7	Saucer
8	Left Thumper Bumper
9	Right Thumper Bumper
10	Center Thumper Bumper
11	Left Slingshot
12	Right Slingshot
13	Drop Target Reset

Note: The solenoid numbering is completely different than used in the original manual. In particular, the numbers 0 through 4 are unassigned. As well, the Coin Lockout and K1 Flipper Enable Relay are not included in the list. The reasons for this are obscure and technical!

Switches

No.	Switch
0	Drop Target “D” (Bottom)
1	Drop Target “C”
2	Drop Target “B”
3	Drop Target “A” (Top)
4	Shooter Lane Rollover
5	Game Button
6	Tilt
7	Out Hole
8	Coin 3 (Right)
9	Coin 1 (Left)
10	Coin 2 (Center)
11	
12	
13	
14	
15	Slam
16	
17	Lower Hyperspace Rollover
18	Right Flipper Lane
19	Left Flipper Lane
20	500 Points
21	Upper Hyperspace Rollover
22	10 Point Rebound (4)
23	300 / 3000 Points
24	Planetary Bypass Inside Lane
25	Planetary Bypass Outside Lane
26	Klingon Target “Y”
27	Klingon Target “L” #2
28	Klingon Target “L” #1
29	“A” Lane
30	“B” Lane
31	Saucer
32	
33	Right Out Lane
34	Left Out Lane
35	Right Slingshot
36	Left Slingshot
37	Center Thumper Bumper
38	Right Thumper Bumper
39	Left Thumper Bumper

Switch Matrix

0: Drop Target "D" (Bottom)	8: Coin 3 (No Diode!)	16:	24: Planetary Bypass Inside Lane	32:
1: Drop Target "C"	9: Coin 1 (No Diode!)	17: Lower Hyperspace Rollover	25: Planetary Bypass Outside Lane	33: Right Out Lane
2: Drop Target "B"	10: Coin 2 (No Diode!)	18: Right Flipper Lane	26: Klingon Target "Y"	34: Left Out Lane
3: Drop Target "A" (Top)	11:	19: Left Flipper Lane	27: Klingon Target "L" #2	35: Right Slingshot
4: Shooter Lane Rollover	12:	20: 500 Points	28: Klingon Target "L" #1	36: Left Slingshot
5: Credit/Game Button	13:	21: Upper Hyperspace Rollover	29: "A" Lane	37: Center Thumper Bumper
6: Tilt	14:	22: 10 Point Rebound (4)	30: "B" Lane	38: Right Thumper Bumper
7: Outhole	15: Slam (3)	23: 300 / 3000 Points	31: Saucer	39: Left Thumper Bumper

Notes:

1. The coin 1, 2, and 3 switches do not have diodes. This means, if used during a game, they could cause a switch matrix issue. They are also handy for testing and understanding switch matrix issues for this reason.
2. The columns are labeled ST 0 through 4 (ST for strobe), and are wired to the MPU board through connector pins A4J2-1 through 5.
3. The rows are labeled I 0 through 7 (I for input), and are wired to the MPU board through connector pins A4J2-8 through 15.

Sounds

Arduino	Sound	Arduino	Sound
0		30	Kirk: Beam me up Scotty!
1		31	Scotty: Hello Computer
2		32	McCoy: You're out of your Vulcan mind!
3		33	Scotty: The warp drive is a hopeless pile of junk
4		34	Spock: The logic was impeccable Captain
5		35	Spock: Vulcans never bluff
6		36	Kirk: Standard orbit Mr. Chekhov
7		37	
8	Beep 3- 10 Chime	38	
9	Beep 44 - 100 Chime	39	
10	Computer beep - 1000 Chime	40	
11		41	Small explosion 1
12		42	Small Explosion 2
13	Command Sequence 1	43	Small explosion 3
14	Command Sequence 2		
15	Command Sequence 3	…	
16	Command Sequence 4		
17		51	Kirk: Scotty, Energize
18		52	Kirk: James Kirk, commanding the Starship Enterprise
19	Power Up	53	Kirk: Fully Operable Scotty?
20	{Computer: Working...	54	Scotty: Prepare to take us out of orbit
21	Fire all phasers: bonus x 1	55	Uhura: It's a big galaxy, Mr. Scott
22	Fire all phasers: bonus x 2	56	Transporter (Louder)
23	Fire all phasers: bonus x 3	57	
24	One Minute to Auto Destruct	58	
25	Bridge to Captain	59	
26	Fascinating	60	
27	Fool Me Once...	61	Kirk: Ahead warp factor 1
28	If My Granny had wheels...	62	Kirk: Full ahead warp factor 1
29	I Can't Change the Laws of Physics	63	Kirk: Full ahead Mr. Sulu

Sounds Pg 2

Arduino	Sound	Arduino	Sound
64	Mr. Spock full ahead warp factor 1	109	
65	Warp 1 Mr. Sulu	110	
66	Kirk: Arm photon torpedoes	111	
67		112	
68		113	Khan! (Louder)
69			
70		…	
71	Photon torpedoes fire, 3 quick		
72		122	Uhura: Calling Klingon vessel
73		123	Phasers on stun
74		124	Prepare to transport
75		125	Try again
76	We may be here for a very long time	126	
77		127	
78		128	
79		129	
80		130	Sulu: Intruder alert
81	He's intelligent but not experienced	131	Sulu: All hands to battle stations
82	He's dead captain	132	All phasor banks fire
83	He's dead Jim	133	Uhura: Klingon vessel acknowledge please
84	Most illogical	134	Uhura: sensor picking up a Klingon battle cruiser
85	You can't argue with a machine	135	Chekov: Number two shield is gone
		136	Kirk: Beam me aboard
…		137	Kirk: Beam me up Mr. Spock
		138	Kirk: How bad is it
104	Small explosion clipped	139	Kirk: Phasers on stun
105	Small explosion clipped	140	Kirk: Prepare to attack
106	Small explosion clipped	141	Kirk: We're running out of time
107	Khan!	142	Kirk: Activate tractor beam
108	Phaser	143	Kirk: We need power

Sounds Pg 3

Arduino	Sound	Arduino	Sound
144	McCoy: I could cure a rainy day		
145	McCoy: Fascinating	…	
146	McCoy: I'm a doctor not a bricklayer		
147	McCoy: I'm a doctor not a mechanic	180	Bridge background
148	McCoy: Most cold-blooded man I've ever known	181	Bridge background
149	Scotty: Alright you lovelies	182	Brass monkeys background
150	Scotty: We'll get more speed out of her	183	Time reverse background
151	Spock: Any chance these are hallucinations	184	Trouble with Tribbles background
152	Spock: I'm sure there is an answer	185	Silvery orbs background
153	Spock: Engine systems coming on	186	Goodbye Mr. Decker background
154	Spock: Extremely interesting	187	Series main title background
155	Spock: Illogical	188	Opening notes
156	Spock: Irritating one of your earth emotions		
157	Spock: Live long and prosper	…	
158	Spock: Can you manage warp 7		
159	Spock: This is not a drill	202	Powerup Chime
160	Scotty: I'll sit on the warp engines and nurse them	203	{Powerup Tune
161	Kid Mode: It’s a big galaxy, Mr. Scott	204	Coin up Noise
162	Kid Mode: Hello, Computer	205	Coin up Tune
163	Kid Mode: Scotty, energize!	206	{Startup Noise
164	Kid Mode: Fully Operable Scotty? Yes sir.	207	Startup Tune
165	Kid Mode: Warp factor one!	208	
166	Kid Mode: Battle Stations!	209	
167	Kid Mode: USS Enterprise calling Klingon vessel. Acknowledge please.	210	
		211	10 Noise
		212	100 Noise
		213	1000 Noise
		214	Saucer Pre-Tune
		215	Saucer Noise
		216	Bally Bonus Tune
		217	Tilt Tune
		218	Game Over Tune

DIP Switches

The original Bally/Stern MPUs had 32 DIP switches for the purpose of customizing the games. Most, if not all, replacement boards have the same switches. The Arduino reads the settings of these switches, and uses them in similar, if not identical fashion. See the explanations and table below for information on individual switches.

No.	Switch
1	Games per coin (or coins per game) for coin chute #1. Switches 1 – 5.
2	"
3	"
4	"
5	"
6	High score to date award setting. Switches 6 – 7.
7	"
8	Sound option (together with switch 32. Ignored by the Arduino).

9	Games per coin (or coins per game) for coin chute #3. Switches 9 – 13.
10	"
11	"
12	"
13	"
14	Assign award level to playfield specials and scoring thresholds. Switches 14 – 15.
15	"
16	Number of balls per game (3 or 5).

17	Maximum credits. Switches 17 – 19.
18	"
19	"
20	Number of credits to be displayed, yes or no.
21	Match Feature, on or off.
22	Hyperspace starting value, 2,000 or 4,000.
23	Bally special remains lit, yes or no.
24	Center target remains lit or alternates.

25	Games per coin (or coins per game) for coin chute #2. Switches 25 – 28.
26	"
27	"
28	"
29	Bally starts at 10,000 or 25,000.
30	Outlane special, both lanes lit or alternate.
31	Flipper feed lanes, both lit for hyperspace or alternate.
32	Sound option (together with switch 8. Ignored by the Arduino).

Switches 01 – 05, 09 – 13, and 17 – 20: Payment for Games

The first five switches are used together to set the number of credits per coin, or coins per credit, for coins dropped into coin chute #1. The original machines set up 32 different payment schemes. Only one is not reproduced (rows 1 and 2 in the chart below. Both are 3 credits for two coins, with one credit on the first coin and two on the second, unless you score points between the first and second coin, in which case you only get one. It really didn't seem worth the effort!)

Switches 09 – 13 are set up the same, setting the credits per coin for coin chute #3. Switches 25 – 28 are set up slightly differently, for coin chute #2.

No.	**Switches**					**Credits / Coin**
Chute #1	**5**	**4**	**3**	**2**	**1**	**(No credits until**
Chute #3	**13**	**12**	**11**	**10**	**9**	**2nd coin is dropped)**
1	OFF	OFF	OFF	OFF	OFF	3 Credits / 2 Coins
2	OFF	OFF	OFF	OFF	ON	3 Credits / 2 Coins
3	OFF	OFF	OFF	ON	OFF	1 Credit / 1 Coin
4	OFF	OFF	OFF	ON	ON	1 Credits / 2 Coins
5	OFF	OFF	ON	OFF	OFF	2 Credits / 1 Coin
6	OFF	OFF	ON	OFF	ON	2 Credits / 2 Coins
7	OFF	OFF	ON	ON	OFF	3 Credits / 1 Coin
8	OFF	OFF	ON	ON	ON	3 Credits / 2 Coins
9	OFF	ON	OFF	OFF	OFF	4 Credits / 1 Coin
10	OFF	ON	OFF	OFF	ON	4 Credits / 2 Coins
11	OFF	ON	OFF	ON	OFF	5 Credits / 1 Coin
12	OFF	ON	OFF	ON	ON	5 Credits / 2 Coins
13	OFF	ON	ON	OFF	OFF	6 Credits / 1 Coin
14	OFF	ON	ON	OFF	ON	6 Credits / 2 Coins
15	OFF	ON	ON	ON	OFF	7 Credits / 1 Coin
16	OFF	ON	ON	ON	ON	7 Credits / 2 Coins
17	ON	OFF	OFF	OFF	OFF	8 Credits / 1 Coin
18	ON	OFF	OFF	OFF	ON	8 Credits / 2 Coins
19	ON	OFF	OFF	ON	OFF	9 Credits / 1 Coin
20	ON	OFF	OFF	ON	ON	9 Credits / 2 Coins
21	ON	OFF	ON	OFF	OFF	10 Credits / 1 Coin
22	ON	OFF	ON	OFF	ON	10 Credits / 2 Coins
23	ON	OFF	ON	ON	OFF	11 Credits / 1 Coin
24	ON	OFF	ON	ON	ON	11 Credits / 2 Coins
25	ON	ON	OFF	OFF	OFF	12 Credits / 1 Coin
26	ON	ON	OFF	OFF	ON	12 Credits / 2 Coins
27	ON	ON	OFF	ON	OFF	13 Credits / 1 Coin
28	ON	ON	OFF	ON	ON	13 Credits / 2 Coins
29	ON	ON	ON	OFF	OFF	14 Credits / 1 Coin
30	ON	ON	ON	OFF	ON	14 Credits / 2 Coins
31	ON	ON	ON	ON	OFF	15 Credits / 1 Coin
32	ON	ON	ON	ON	ON	15 Credits / 2 Coins

No.	Switches				Credits / Coin
Chute #2	28	27	26	25	
1	OFF	OFF	OFF	OFF	Same as chute #1
2	OFF	OFF	OFF	ON	1 Credit / 1 Coin
3	OFF	OFF	ON	OFF	2 Credits / 1 Coin
4	OFF	OFF	ON	ON	3 Credits / 1 Coin
5	OFF	ON	OFF	OFF	4 Credits / 1 Coin
6	OFF	ON	OFF	ON	5 Credits / 1 Coin
7	OFF	ON	ON	OFF	6 Credits / 1 Coin
8	OFF	ON	ON	ON	7 Credits / 1 Coin
9	ON	OFF	OFF	OFF	8 Credits / 1 Coin
10	ON	OFF	OFF	ON	9 Credits / 1 Coin
11	ON	OFF	ON	OFF	10 Credits / 1 Coin
12	ON	OFF	ON	ON	11 Credits / 1 Coin
13	ON	ON	OFF	OFF	12 Credits / 1 Coin
14	ON	ON	OFF	ON	13 Credits / 1 Coin
15	ON	ON	ON	OFF	14 Credits / 1 Coin
16	ON	ON	ON	ON	15 Credits / 1 Coin

Switches 06 and 07: High Score Award

Award for Beating High Score	Switch 7	Switch 6
No Award	OFF	OFF
One Credit	OFF	ON
Two Credits	ON	OFF
3 Credits	ON	ON

Switch 08: Sound Options

Switches 8 and 32 originally worked together to produce a cacophony of electronic tunes, buzzes, chimes, and knockers. These two switches are ignored.

Switches 14 and 15: Playfield and Scoring Threshold Special Awards

Switch 14	**ON**	**OFF**	**OFF**
Switch 15	**ON**	**ON**	**OFF**
Outlane Special	Replay	X-Ball*	50,000
Drop Target Special	Replay	X-Ball*	50,000
B-A-L-L-Y Special	Replay	X-Ball*	50,000
Hyperspace Special	X-Ball	X-Ball**	25,000
Scoring Thresholds	Replay	X-Ball**	No Award

* 50,000 awarded if Same Player Shoots Again already lit
** 25,000 awarded if Same Player Shoots Again already lit

Switch 16: Balls Per Game

Liberal	ON	5 balls per game
Conservative	OFF	3 balls per game

Switches 17 - 19: Maximum credits allowed (as per original Bally manual):

Maximum	Switches		
Credits	**19**	**18**	**17**
5	OFF	OFF	OFF
10	OFF	OFF	ON
15	OFF	ON	OFF
20	OFF	ON	ON
25	ON	OFF	OFF
30	ON	OFF	ON
35	ON	ON	OFF
40	ON	ON	ON

Switch 20: Credits displayed, YES or NO
Switch 21: End of game match feature, ON or OFF

Switch 22: Hyperspace Adjustment

Liberal	ON	Hyperspace begins at 4,000
Conservative	OFF	Hyperspace begins at 2,000

Switch 23: Bally Special Adjustment

Liberal	ON	Bally Special remains lit
Conservative	OFF	Bally Special awarded once

Switch 24: Center Target Adjustment

Liberal	ON	Center target remains lit (3,000 points)
Conservative	OFF	Center target alternates (300 / 3,000 points)

Switch 29: Bally 10,000 – 25,000 Adjustment

Liberal	ON	Bally begins at 25,000
Conservative	OFF	Bally begins at 10,000

Switch 30: Outlane Special Adjustment

Liberal	ON	Both lanes lite for special
Conservative	OFF	Special alternates from side to side

Switch 31: Flipper Feed Lanes Adjustment

Liberal	ON	Both lanes lite for Hyperspace value
Conservative	OFF	Hyperspace award alternates between lanes

Switch 32: Sound Options
Switches 8 and 32 originally worked together to produce a cacophony of electronic tunes, buzzes, chimes, and knockers. These two switches are ignored.

Kids' Mode

Kids' mode is a new feature that allows you to easily switch your pinball to all liberal settings, for play by kids or guests. Just press the Game Button and Coin Slot 3 switch at the same time while in either Attract Mode or Self-Test (other than during the Switch Tests). To go back to regular game mode, press the two buttons again. You can also simply turn the pinball off and on, as it will always start in regular game mode. Sound cues will remind you occasionally of the game mode you are in.

Note that the high score in Kids' Mode starts at 100,000, allowing kids to compete for the day. When you return to regular game mode, through self-test or by turning the pinball off and on, the Kids' Mode high score will be gone, and the original high score as well as all the original settings will be reinstated.

Kids' Mode resets all the settings in the tables below:

DIP Switches
Hyperspace begins at 4,000 points
B-A-L-L-Y Special remains lit for extra ball
300 / 3000 Target remains lit
B-A-L-L-Y begins at 25,000 points
Both Out Lanes remain lit for extra ball
Both Flipper Lanes remain lit to score Hyperspace value
5 balls per game

Self-Test Game Settings	**Value**
High Score	50,000
Score Award 1 (Extra ball)	25,000
Score Award 2 (Klingon Battle mini-game)	75,000
Score Award 3 (Auto-Destruct mini game)	150,000
Personal Goal	75,000
Playfield / Threshold Award	2 (Extra Ball)
Ball Saves Allowed	5

Programmable Game Adjustments	**Value**
None	

User Programmable Changes*

There is a section near the top of the main program titled “Operator Game Adjustments”, which can be used to adjust features of the game. Most of the really useful items that were ever included here have been converted to Self-Test adjustments. The following items remain. If you are familiar with programming and are able to compile the software to your Arduino, you may want to look at these potential changes:

ATTRACT_SPEECH_TIMER: Time between attract mode callouts. Usually 5 minutes.
MAX_TILT_WARNINGS: Usually 1. Set higher if you wish to allow some number of tilts before ending a ball.

VERSION_NUMBER: The program expects a version number in the format yyyy.mm, representing the year and month of your changes. This version number is displayed when the pinball is turned on. If you are modifying the program, you should likely copy the software to a new directory named STyyyypmm, and rename the two files that use this naming format. Then change this version number in the software to make clear which version is running.

DEBUG_MODE: There are a number of programmable features available to assist you in identifying and possibly eliminating some common pinball issues. Currently there are three debug modes, available by setting DEBUG_MODE to one of the following values:

0. Set to zero to indicate no debug messages are required (regular game play).
1. Monitor the switches as they are hit. The most recent nine switches are scrolled through the player 2, 3, and 4 displays.
2. Display the number of times per second the switches are monitored. Player 4 display is used. The result is highly game and activity-specific. The result for the Star Trek is usually around 1000 – 1200, although for more complex games the result is often in the range of about 180 to 200 times a second.

By playing a single player game with debug mode 1 set, you can monitor the switches being activated in real time. This can be very useful if you are experiencing switch issues. Note that it must be a single player game, as the player 2, 3, and 4 displays are used to display the switch values. Each 6-digit display will show three two-digit switch numbers. The most recent switch is displayed in the last two digits of Player 4, and will scroll left and up as new switches are hit.

Pressing and holding the game credit button will stop the monitoring of new switch hits, for as long as you hold the button. This allows you to stop the numbers from scrolling away, so you can review them more closely.

Debug mode can be used to track down issues with switches misfiring. Depending on the problem, the following may be helpful in eliminating the issue. Try cleaning and re-gapping the switches first, but if this fails to resolve the issue you may want to try the following:

* This section assumes some minimal programming skill on your part. If you are uncomfortable with making changes to the program, maybe skip this section! Always save a copy of the program before making changes.

SwitchDebounce[]: This variable can be found near the top of the main program, shortly following the Operator Game Adjustments. It contains 40 rows, one for each switch, and two columns. It is used in monitoring the switches for multiple hits. Occasionally, a switch can develop a "bounce", where it registers two or more times for a single hit by a pinball. By setting a value from 0 to 255 in the second column, the pinball will ignore any hits on that switch for that number of milliseconds following an initial hit.

For example, I was having trouble with the left outlane rollover (switch 34) registering multiple times whenever it was hit. To fix this I set column 2 of row 34 to 250. So now, after the switch is hit it will not register another hit for at least 250 milliseconds.

The Switch Bounce Test can be used to test switches for bounce. See the Switch Bounce Test documentation in the Self-Test section of this document. This test will also tell you the time between double-hits, which can be used as a minimum value in setting the second column of this variable.

ResetHits[]: This variable can also be found near the top of the main program, right after the SwitchDebounce[] variable. It contains 14 rows, one for each solenoid, and 2 columns. It is used to eliminate switches from activating due to vibration from solenoids firing. Similar to SwitchDebounce[], a value from zero to 255 can be set in the second column, causing the pinball to ignore hits to specific switches for the indicated number of milliseconds following a solenoid firing.

To make this work, you also need to indicate which switches are to be ignored following which solenoid firing. This is done by adding code to the function ResetHitFix(). There are examples of the required code in the function already, which you can use to model your own.

For example, suppose that resetting the drop targets is causing the right outlane switch to activate. You've tried cleaning and re-gapping the switch but can't seem to get it to stop. To fix this you can set row 13 (SO_DTARGET_RESET), column 2 of ResetHits[] to 250. Then add the following code to ResetHitFix():

```
case SW_RIGHT_OUTLANE:
    if (CurrentTime < ResetHits[SO_DTARGET_RESET].start) return false;
    if (CurrentTime - ResetHits[SO_DTARGET_RESET].start
             < ResetHits[SO_DTARGET_RESET].wait) return true;
    return false;
```

This says, if the right outlane target is hit, and a drop target reset has occurred, check how long it has been since the drop target fired. If less that 250 milliseconds ago, the switch hit is to be ignored (return true).

Coding has already been added to prevent any drop target solenoid from setting off its own drop target switches. This can be used as an example to model your own code.

Setting the Wait Time: Both SwitchDebounce[] and ResetHits[] depend on you setting a "wait" time. This is the time between events during which the pinball will ignore specific switches. The wait time must be longer than the time it takes for vibration to set off the second switch hit, but shorter than the time it can take for a switch to be hit legitimately, with a maximum of 255ms.

250ms, or a quarter of a second, is generally a good choice.

The time between a solenoid firing, and a switch being activated by vibration, can be determined in the Solenoid self-test. The time taken for a switch bounce can similarly be determined in the Switch Bounce self-test. The results of these tests would be the absolute minimum value you should use for the wait time, although you likely want something higher.

For example, I had a drop target that would frequently set off a target switch on the other side of the playfield. It would be impossible to activate the drop target solenoid and then the target switch in less than several seconds, much longer than the maximum 255ms. The solenoid test showed that it took about 133ms between the drop target reset until the switch would activate. The wait time could then be an absolute minimum of 133, up to a maximum of 255. I therefore set the wait time to 250, allowing extra time over and above the absolute minimum.

Generally, a longer time is better, since the wait time could be longer than that given by the Solenoid or Switch Bounce test. Where it could be more difficult is when the solenoid and the target switch are close together. For example, a pop bumper that can shoot the ball directly into a switch. If the pop bumper is sometimes setting off the switch without hitting it, you will need to determine not only the minimum time from the solenoid test, but also a maximum time based on how long it can take the ball to travel from the pop bumper to the switch. If this is less than a quarter second, you may need to do some careful timing calculations.

Other Programming Changes

Kids' Mode: Kids' Mode is described above, and is meant to allow you to convert the game to easy settings, primarily for kids to play. However, another use of this might be to allow players with different setting preferences to share a pinball machine. For example, maybe you like a five-ball game, but your brother insists on three. In this case, you might not want ALL the settings at easy levels, only the number of balls.

This would be easy to do. There is a function KidSettings() in the main program file. By commenting out or rewriting the lines in this function, Kids' Mode can be adapted to the alternative game settings you prefer. Comparing the variable names in the software to the items in Self-Test section 24 should make it obvious what changes need to be made.

Sound Changes: This program was not written with replacing the sounds in mind! However, it is possible. Just, not easy. The following suggestions should help. Search the internet for "Robertsonics WAV Trigger" for additional requirements and suggestions when creating your own sound files. The Audacity audio editor is recommended. 44100, 16-bit, stereo, with all metadata deleted, is required.

Sound files can be changed on the WAV Trigger's micro-SD card. Notice that every file begins with a 4-digit number between 0001 and 0255. The WAV Trigger looks at this number, and ignores the remainder of the file name. By replacing any sound file with a file of your own that begins with the same number, the game will play your file instead. For example, to change the background music, you could make up your own files 0181 – 0187. You should also review the variable SoundTimings, near the bottom of file ST2025p04.h. This records the sound type (background, voice, or sound effect) and length (in 10ths of a second) for each file (background music files should have length set to zero). Note that files with sound type zero will not play!

Changing more than a very few files this way could get tedious. If you want to change most or all of the files, I would create a new directory on your computer. Create all your new sound files in this directory. Assign them all numbers from 0001 to 0255, and add these numbers to the beginning of each sound file. Don't worry about matching my numbers.

You will then need to rewrite the SoundTimings variable (near the end of ST2025p04.h), assigning sound types and lengths for all your sounds.

In ST2025p04.h, below the SoundTimings variable, are a number of variables set up for the purpose of assigning sounds to events in the game. Some of these are lists of sound file numbers, and the event will choose from the list randomly. For example: VoiceTilt[4] = {3, 27, 33, 76};

is used to select one of three voices to be played when the game is tilted. The first number (3) must be the number of sound files in the group, followed by that many sound files (in this case, 27, 33, and 76).

Others are single values, for example: VoiceBallSave = 125;

indicates that sound file 125 will be played when ball save is applied ("Try again").

Once you have made up your directory of sound files, go through these variables, deleting my numbers where you want and replacing them with yours. Save the old files from the micro-SD card in another directory, and replace them with the new files from your directory. Finally, recompile the program to your Arduino. I strongly recommend you keep a copy of your new ST2025p04.h file with the sound files, as they are required to work together. A directory of sound files played with an incorrect SoundTimings or assignment variables will not work properly.

I warned you it wasn't going to be easy!

Notes

DIP Switches: The DIP switches for the Arduino have been set up to be as identical to the switches for the regular game as possible. So simply installing the Arduino in your pinball should have it run according to the rules you are familiar with. See the original Bally Star Trek manual for more information on the DIP switches, and on the self-tests and game settings.

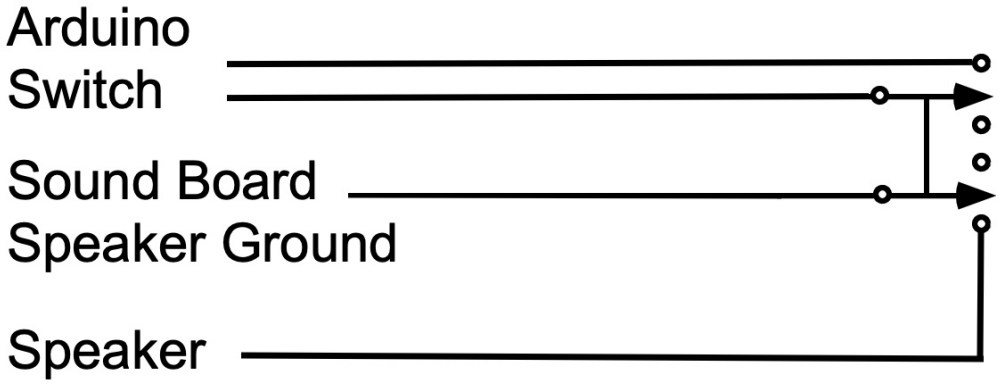
A new test has been added to the self-tests, to display and change the DIP switch settings easily, but temporarily. See the Self-Test section above for more information.

Self-Test Settings: The original self-test settings, set on your Star Trek MPU, are not used by the Arduino. The first thing you should do after installing the Arduino and before playing a game is to enter Self-Test, and update all the game settings to the values you desire.

Kids' Mode: A new feature has been added known as Kids' Mode, that changes all game settings to their easiest settings temporarily. See the Game Settings section above for more information.

Switch to Original: It is possible to add a switch to the Arduino, on long wires running from the 2-pin connector labeled "Switch", and then out the air vents at the back, so that you will be able to easily switch back and forth between the new and old rules. But don't bother. You are never going to want to play the old rules again!

If you do want to be able to play the old rules, keep in mind that the WAV Trigger board will not play the old sounds. You will need to set up another switch to control where the sound is directed. I would recommend a double-pole, double-throw switch, which allows you to control both with a single switch as below:

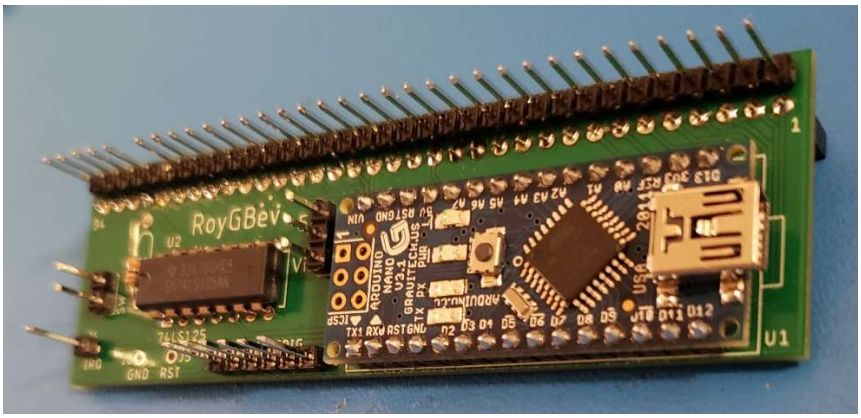


When the switch is up, the Arduino switch is connected and the sound board speaker is disconnected, so game and sound are controlled by the Arduino. When the switch is down, The Arduino switch is disconnected and the sound board is connected to the pinball's original speaker, so the original game is running and playing the original sound through the original speaker.

Bally / Stern Arduino pinball adapter installation instructions *

Part Description	Qty
Bally/Stern Arduino adapter PCB	1
Push-release IRQ wire harness	1
(Optional) .wav trigger sound board	1
(Optional) 6-pin ribbon cable 1	1

The Arduino adapter in your kit will be one of two types: The REV 1 board (Figure 1), or the REV 3 board (Figure 2).



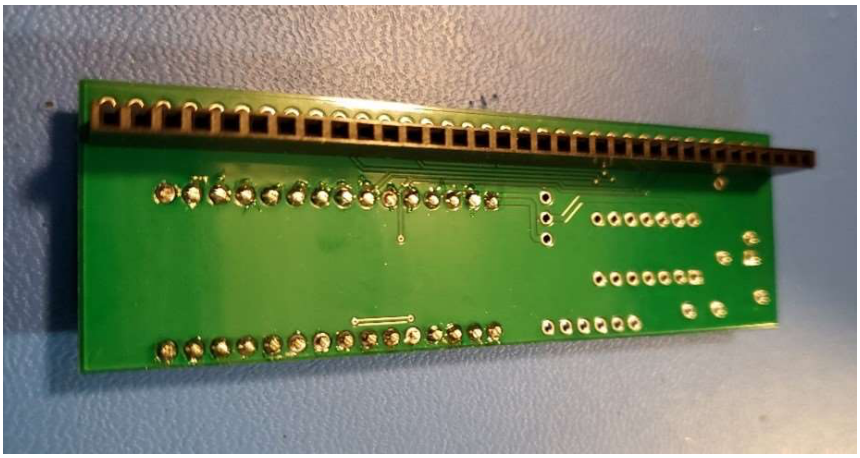


Figure 1

* These are the complete installation notes provided with an Arduino Mega 2560 REV 3. They are unedited, and so may contain extra information not needed in installing your Star Trek Arduino. It also includes full instructions for installing and connecting a WAV Trigger sound board. Additional notes of my own are included at the end.

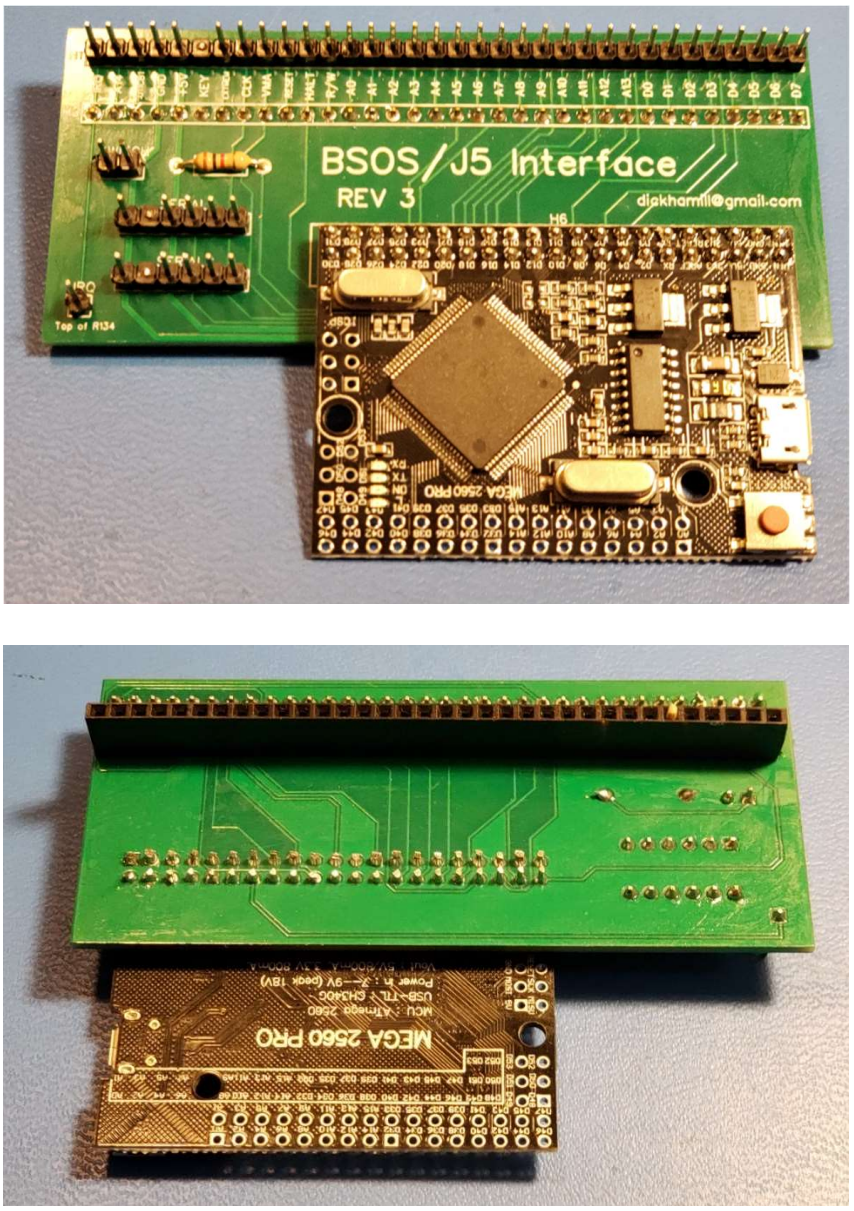


Figure 2

IMPORTANT: TURN OFF POWER TO THE GAME BEFORE INSTALLING THE ARDUINO ADAPTER.

1. Verify that the game's MPU board is in good working condition. Verify that the game powers up and works.

2. If the game has a ribbon cable connecting a sound board to J5 on the MPU, note the orientation of the cable, then remove it.

3. Verify that the MPU connector J5 is in good condition, with good solder joints. Also verify that key pin 29 of J5 has been removed. J5 pins are numbered from pin 1 on the right.

4. If your kit uses the REV 1 Arduino board, install a 0.1" jumper between the Vin and middle pins of the 3-pin header, as shown in Figure 3. The pin labeled 5V should be unconnected.

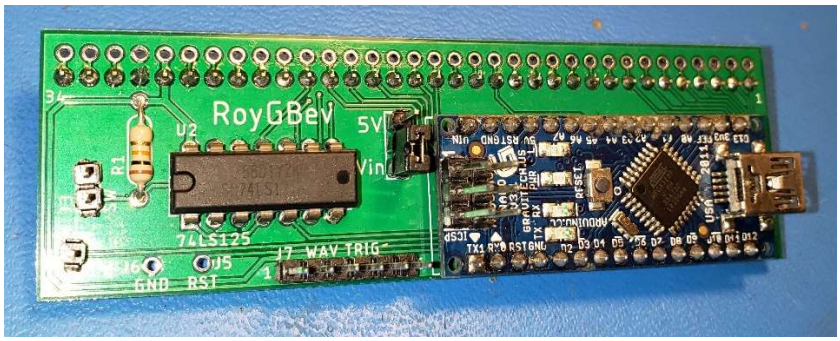


Figure 3

5. The 34-pin female connector on the bottom of the adapter will be plugged into J5 on the game's MPU board. If you are using an Alltek or Stern MPU-200 board, J5 will have 34 pins. On other boards, J5 may have 32 or 33 pins. In all cases, pin 1 on the adapter will align with pin 1 on J5, and the blocking pin at pin 29 of the adapter will align with the missing pin 29 on J5.

6. Line up pin 1 and carefully install the adapter board onto connector J5. If the connector won't go, don't force it. Verify that pin 1 is lined up, and that there are no bent pins preventing engagement. Push the adapter down onto J5 until it is fully installed. Figure 4 shows an adapter installed on an MPU board.

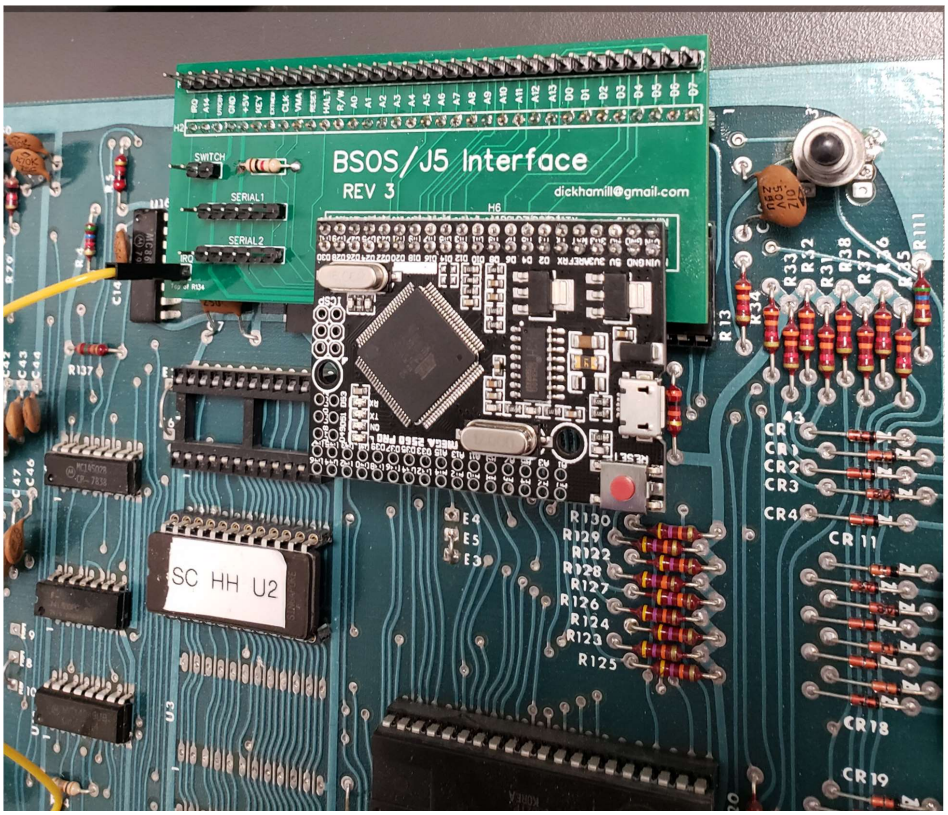


Figure 4

7. If a sound board ribbon cable was removed from J5 in step 2, install the cable onto the 34-pin connector on the Arduino adapter.

8. If your MPU board is a Bally AS-2518-17, Bally AS-2518-35, or Stern MPU-100, the push-release IRQ wire harness will need to be installed. To install:

   a) Push the single pin female connector onto the pin labeled IRQ on the Arduino adapter.
   b) Connect the push-release probe to the top lead of R134 on the MPU board. Figure 5 shows this in detail.

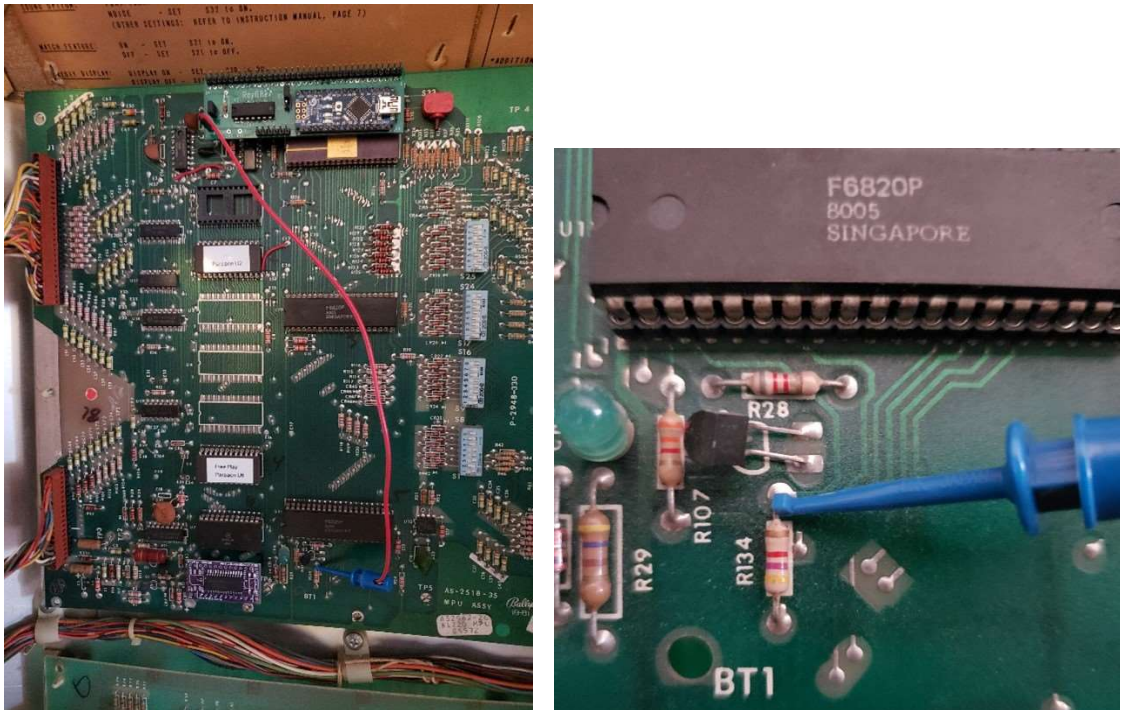


Figure 5

9. The two-pin header labeled SWITCH on the adapter board should not have a jumper installed. Verify this, and remove the jumper if it is installed.

10. At this point the game should be ready to power up. Watch the diagnostic LED on the MPU board while powering up the game. The game should power up normally, to the original 6800 game program, and into attract mode. If it does not, turn the game off and check the installation. If a sound board ribbon cable was attached to the Arduino adapter's 34-pin male header, unplug the cable, then power up the game again. If it powers up normally, there is an issue with the sound board ribbon cable installation.

11. Turn off the game, then install a jumper between the two pins of the connector labeled SWITCH. IMPORTANT: THE GAME MUST BE TURNED OFF BEFORE THE SWITCH JUMPER IS INSTALLED OR REMOVED.

12. Turn on the game. It should boot up to the new Arduino code. Check that the displays show digits, and that the game enters attract mode.

13. If the game does not boot to the Arduino software version, turn off the game and check that the IRQ wire harness is properly installed.

14. If desired, a remote switch may be connected to the pins of the SWITCH connector, allowing the user to select between standard and Arduino versions. As with the jumper, TURN OFF THE GAME BEFORE CHANGING THE SWITCH SETTING.

15. The new Arduino software may support configuration settings that can be selected using the game's self-test switch inside the coin door. Refer to the user documentation for the Arduino pinball software for details.

16. If the Arduino installation kit includes a Robertsonics .wav trigger sound board, refer to the following steps for installation.

17. TURN OFF POWER to the game before proceeding.

18. Refer to Figure 6 for the next few steps.

19. Verify that the Load/Run switch on the sound board is in the Run position.

20. The sound board has a Micro SD card socket with a memory card installed in it. Confirm that the memory card is installed in the socket.

21. Install one end of the 6-pin ribbon cable to the 6-pin connector labeled SERIAL1 (REV 3 board) or J7 WAV TRIG (REV 1 board). The cable has a black stripe on the connector, designating pin 2. Install the cable with the black stripe lined up with pin 2 (the missing pin) on the 6-pin connector.

22. Connect the loose end of the 6-pin ribbon cable to the male header on the sound board. Note that the ribbon cable's connector has a black stripe designating pin 2 on the connector. Line up the black stripe with pin 2 (the missing pin) on the sound board's connector.

23. The sound board has a piece of foam mounting tape attached to the back. This tape can be used to temporarily mount the sound board. Place the sound board in a location close enough that the 6-pin ribbon cable can easily reach. Refer to Figure 7 for an example. For a more permanent installation, the use of mounting screws and standoffs is recommended.

24. The sound board has a 3.5 mm stereo jack for use with speakers. Connect a pair of wired powered speakers, or other wired device (earbuds / headphones) to the jack.

25. To test that the sound board is working, turn the game on. The sound board should play a power up sound when the Arduino game code boots up.

26. If the sound board does not play a sound at power up, test the sound board by pressing the test button. The sound board should make a sound. If it does, but there is no sound during gameplay, the game software may not be configured to use the sound board. Refer to the user documentation for the Arduino pinball software for details.

27.

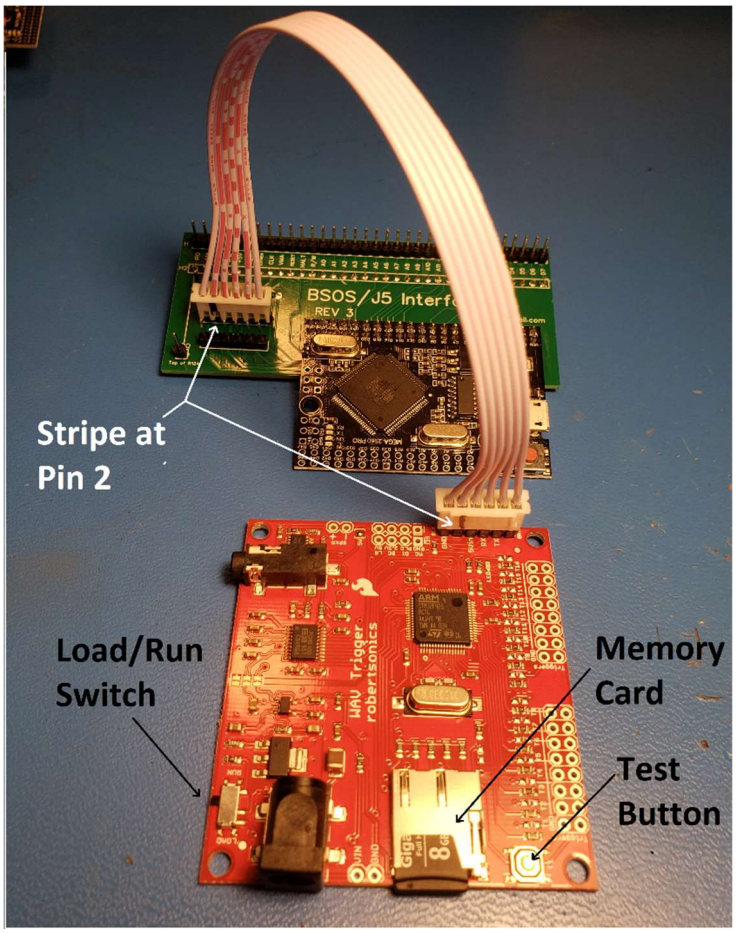


Figure 6

28. Suggested installation: Figures 7 and 8 show an installation for Stern Trident, with powered speakers mounted on top of the game's backbox.

29. Refer any questions regarding this document and installation to roygbevdotcom@gmail.com.

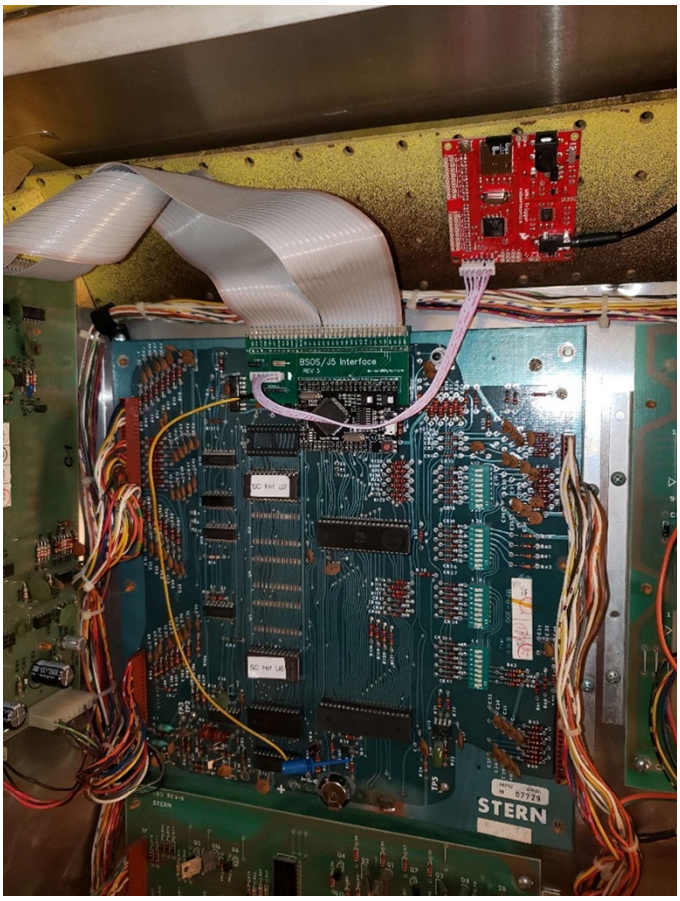


Figure 7

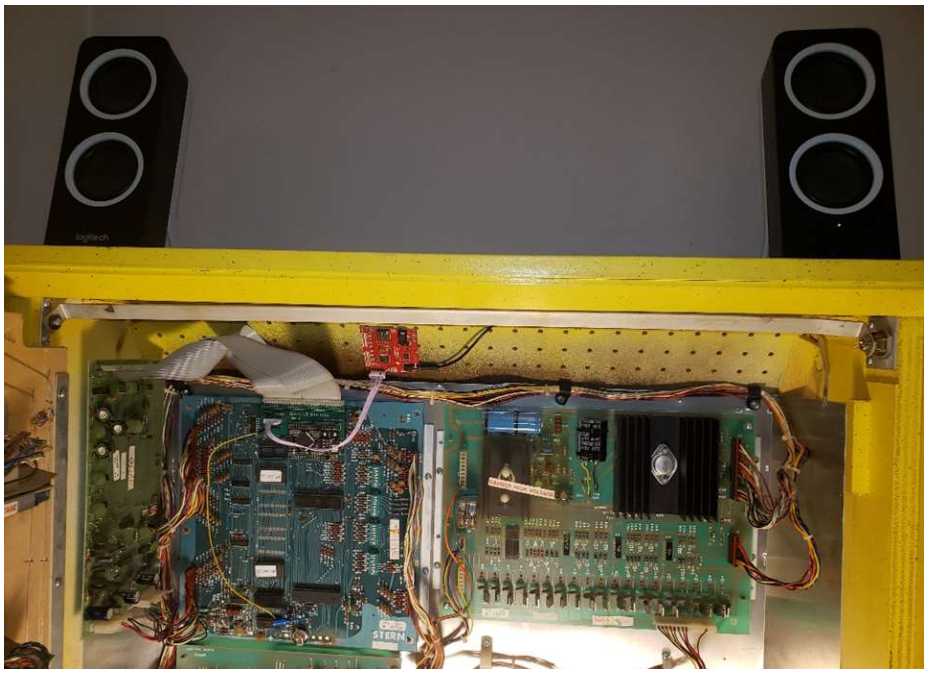


Figure 8

Additional from Dave's Think Tank:

I recommend you purchase a stereo computer speaker bar, to be used as the speakers in the above setup. They cost about $20, so not expensive. It should have a 3.5mm stereo jack to connect to the WAV Trigger, as well as a USB connector to power the speaker. Whatever speakers you use, the WAV Trigger does not power the speakers, so external power is required. The USB connector on the computer speaker can be plugged into an old phone charger to complete the setup.

If you want the speakers to turn off when you turn off the pinball, it is possible to power a small speaker bar (around 1 amp usually) using a 5-volt source in your pinball. This may require some modifications to your voltage regulator board. The method I used is documented below.

Powering a Speaker Using 5-Volts from the Regulator Board

Have you ever noticed on the J3 connector for the voltage regulator board, there is a loop of wire that connects pin 13 to pin 25 (see Figure 9)? This takes 5 volts from one section of the board to another. But if we were to connect those sections directly, this loop could be removed, freeing up two sources of 5 volts which could then be used to power our speaker.

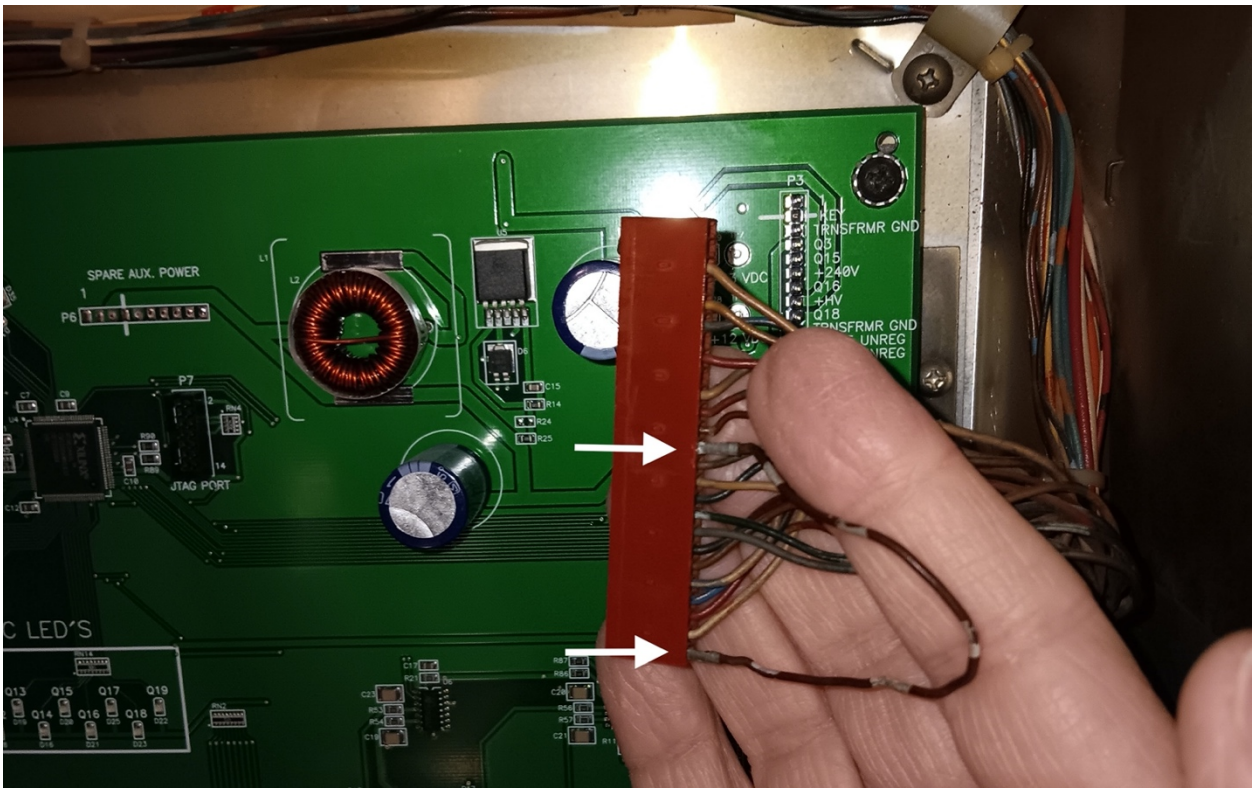


Figure 9

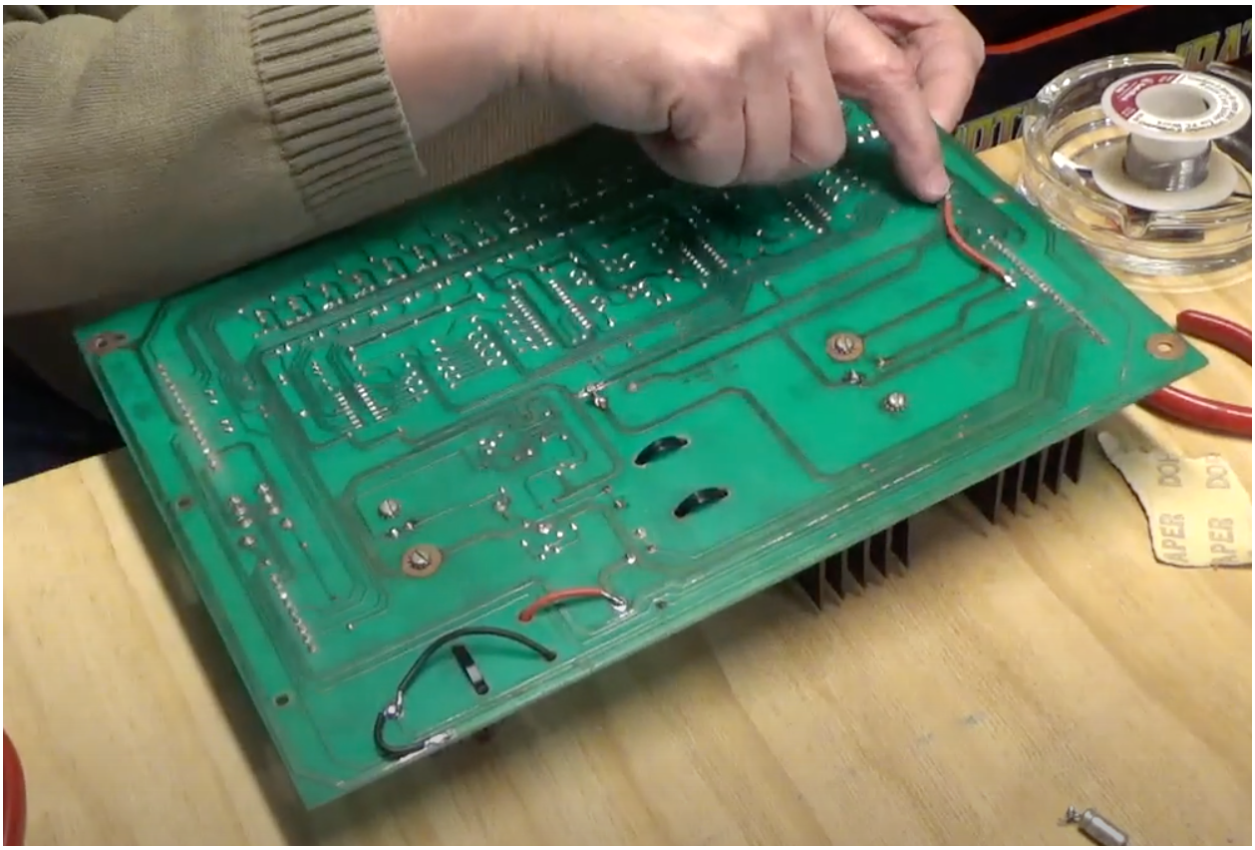


Figure 10

If you have an Alltek voltage regulator board, you do not need to make any modifications. Just remove the looping wire, and you're ready to connect your speaker. But if you are still using the original board, you'll have to make this modification. In Figure 10 you can see I have soldered a wire on the back of the voltage regulator board, connecting TP1 to TP3. This is a recommended modification in any case. TP1 and TP3 are directly connected to J3 pins 13 and 25, and so the loop connecting these pins can be removed.

See this video for additional information: https://www.youtube.com/watch?v=VWA-4wWTFIQ

Next, you need to make a special connector (see Figure 11). This is known as a USB Female Pigtail connector. It has a regular USB female plug on one end, and bare wires on the other. The red wire is your 5-volt line. Crimp a standard 0.1 inch connector to the red wire. The black wire is your ground. There are no spare ground connectors on J3, but we can connect this wire to ground at the corner of the voltage regulator board. Solder a metal washer to the black wire.

There may be two additional wires, that are usually green and white. These are data lines, and not needed for this purpose. Clip them off or tie them back out of the way.

In Figure 12, you can see the connector installed. The red wire is connected into pin 25 of J3, and the soldered washer is held under the corner screw of the board. I have placed it on top of the board, but you may need to place it under the board to get a good connection. Notice also that I have secured the connector's wire using a cable clamp, to eliminate any strain on the connections.

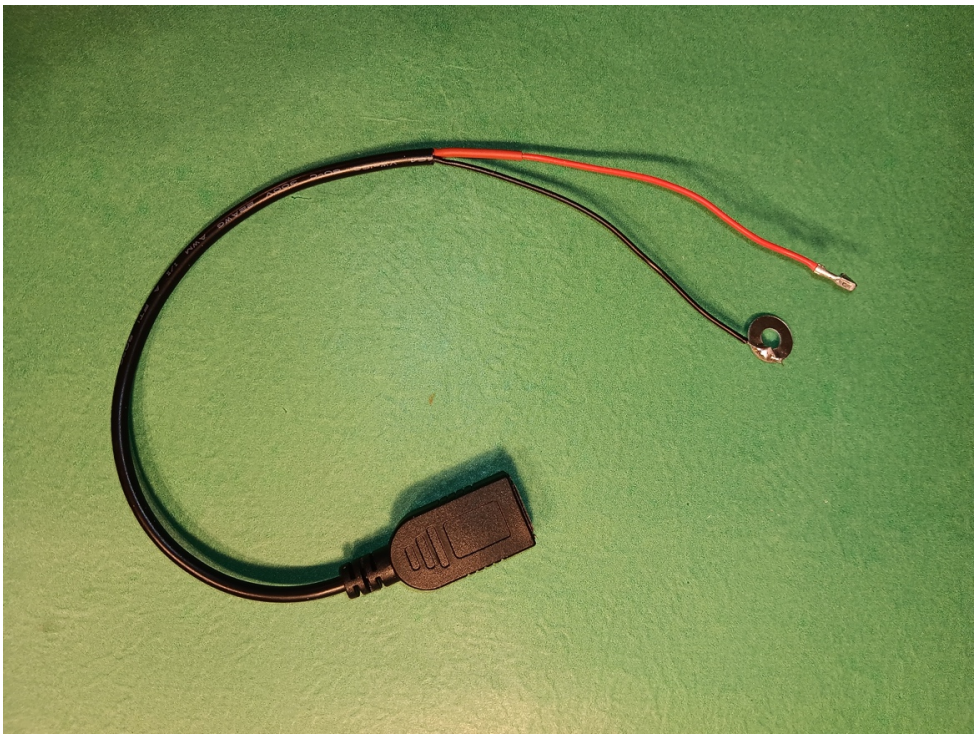


Figure 11

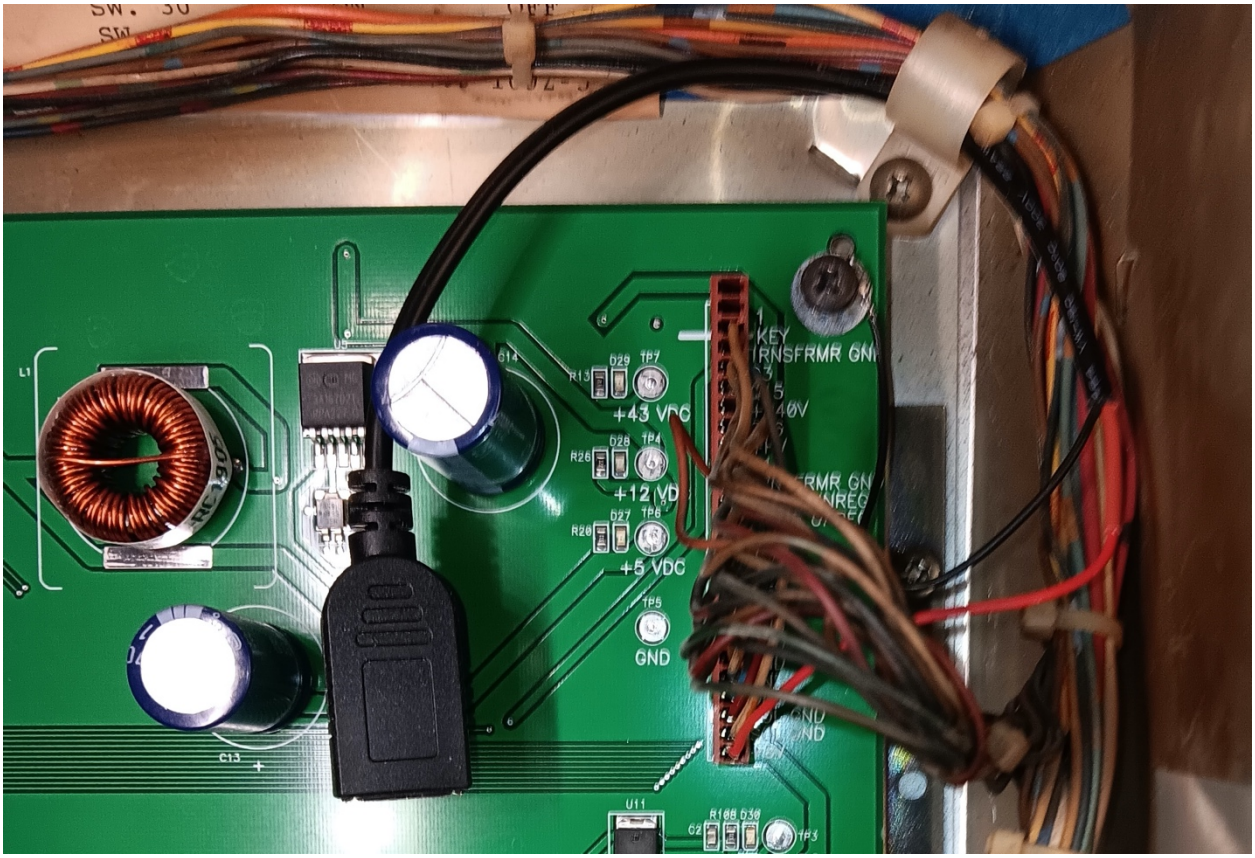


Figure 12

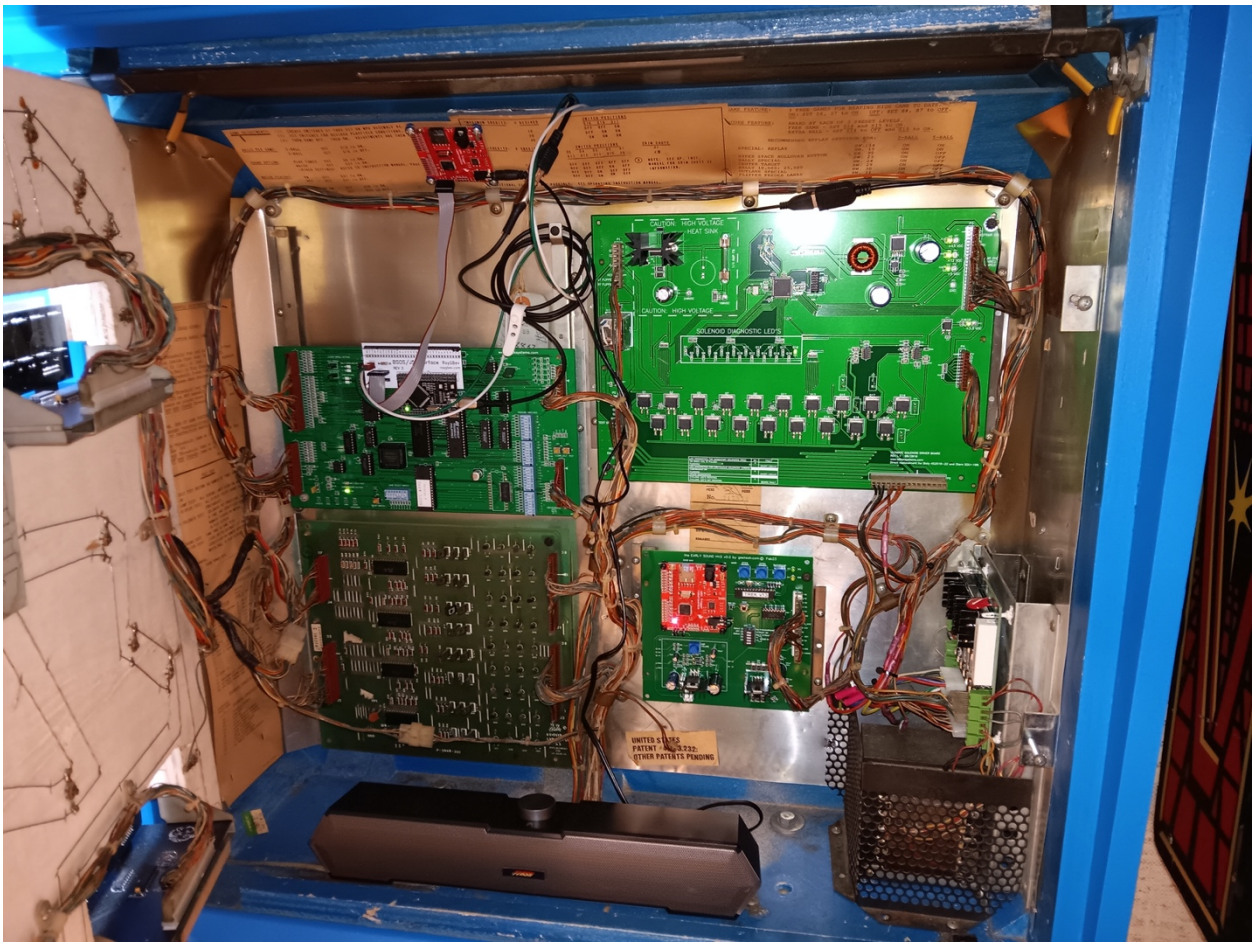


Figure 13

Figure 13 shows the entire setup, including the Arduino board on the MPU, connected to the WAV Trigger board at the top of the backbox, with the speaker bar at the bottom plugged into the WAV Trigger with a 3.5mm stereo jack (bottom right of the board), and plugged into our pigtail connector with a USB connector for power (above the voltage regulator board).

You will likely not want the speaker to be inside the backbox! It is a simple task at this point, though, to move the speaker bar to the top of the backbox, and snake the wires through the vents at the back, into the backbox, to make the necessary connections.

Keen observers will note that the speaker wire from my original sound board is disconnected. This is necessary to prevent this board from playing unwanted sound during games. I have since added a switch on the back of the pinball that connects or disconnects the ground wire from this sound board to the original speaker. In fact, it is the same switch I use to turn the Arduino off and on (a 6-pin, 2-position toggle switch). With the switch down, the Arduino is disconnected and the old speaker is connected. I can then play the original game with the original sound from the original speaker. Or with the switch up, The Arduino board controls the game with sound from the speaker bar, and the old speaker is disconnected (see Figure 14).

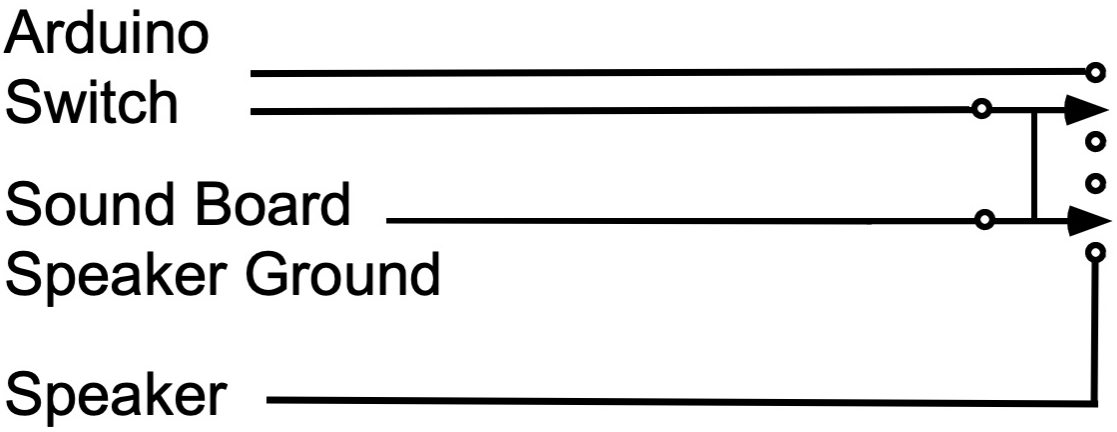


Figure 14