

Arduino TITO and Player Tracking Project

Marc R. Davis

PROJECT GOALS

To allow home slot machine owners the ability to add Ticket In/Ticket Out (TITO), Remote Control, Monitoring and Player Tracking (Display/Keypad/Reader) to their SAS-Compatible games using an Arduino Mega 2560 and reusing existing player tracking hardware

Portions of the Arduino SAS protocol implementation by Ian Walker

Additional testing and troubleshooting by NLG member Eddiie

HARDWARE REQUIRED

Arduino Mega 2560 R3

<https://www.amazon.com/gp/product/B01H4ZLZLQ/>

RFID RC 522 (not needed if you will be using a compatible magnetic reader)

<https://www.amazon.com/gp/product/B07KGBJ9VG/>

W5100 Ethernet Shield

<https://www.amazon.com/gp/product/B00AXVX5D0/>

MAX3232 Serial port (or compatible)

<https://www.amazon.com/gp/product/B083L99CGZ>

6' USB Cable

<https://www.amazon.com/dp/B00NH11KIK>

6' M-F 9-pin straight-through serial cable (some setups may need a shielded and chassis-grounded cable due to electrical interference from game)

<https://www.amazon.com/dp/B006W3XVZK>

Micro-SD memory card – 32GB or smaller

<https://www.amazon.com/gp/product/B073K14CVB>

Compatible slot machine

- Tested on IGT S2000/GameKings/AVP, Bally, WMS and Konami machines; will probably work on others based on the SAS 6.x protocol; you will need a compatible cable to connect to the serial port on the machine

Player tracking bracket with an existing card reader, compatible VF (or LCD) display and a compatible keypad

Compatible Vacuum Fluorescent Display or LCD

- IEE Compatible VF Display such as: IEE 03601-95A-40 using included modified library
- DataVision DV-16236 displays using the LiquidCrystal library
- Noritake GU-7000 Series displays using included modified library
- Noritake CU20025ECPB-U1J display using the LiquidCrystal library in 8 bit mode
- 1602A LCD displays using the LiquidCrystal library (Default)

Compatible Keypad (Optional)

- Bally Systems Player Tracking 6x2 Keypad P/N 105123F (Default)
- Bally Systems Player Tracking 3x4 Keypad
- ACT 8x2 Keypad

Compatible Card Reader (if using Magnetic Stripe cards)

- UIC MSR240-02TMRNWWBR (Figure 3)
- Neuron Card Reader (Not working yet - but similar pinout)

Dupont M-F jumper wires (quantity and length depends on configuration)

CAT5 cable long enough to reach from your network jack to the top box (or use a WiFi-RJ45 dongle)

<https://www.amazon.com//dp/B07QXNWRWJ>

Serial cable to connect to game

- For IGT Games: 5-pin Dupont to Male DB9 Serial pigtail (I reused the one from the BETTORSlots TITO board; I have not found a source for these yet – see Figure 1)

Compatible Player Cards (at least 3)

- RFID: <https://www.amazon.com/gp/product/B01FR66KWJ>
- MAGSTRIPE: Magnetic Stripe cards with Track 2 data
 - o I tested this with a variety of casino tracking cards and other loyalty cards
 - o The program reads the serial number from track 2, drops any leading or trailing zeros and uses the last 8 digits so as to be compatible with the 8.3 filename format limitation; because of this there is a slim chance that two random cards could have the same number

5 Watt or greater USB Power Brick

3D printed case for Arduino (or something to protect Arduino from shorting-out on metal in upper cabinet)

- <https://www.thingiverse.com/thing:30270>
- <https://www.thingiverse.com/thing:1364105>

SOFTWARE REQUIRED

Latest version of Arduino IDE

<https://www.arduino.cc/en/software>

BEFORE YOU BEGIN

These instructions assume familiarity with electronics, coding and slot machine configurations. If you are new to any of these then this project may not be for you. My instructions were not written for beginners. You may damage your game or the Arduino/TITO hardware if you do not understand what you are doing.

PICK YOUR VERSION

There are now two versions of the sketch included in the package – one is for RFID and the other is for Magnetic Stripe readers. Determine which you want to use and modify the file as needed for your display and keypad options.

WIRING

- The Ethernet Shield attaches to the Arduino Mega via the built-in pin headers
- Display Wiring
 - o If using an IEE VFD (assumes IEE Vacuum Fluorescent Display 03601-95A-40) then wire the connections as follows:

IEE VFD	Arduino
Pin 1	Data Pin 22
Pin 2	Data Pin 23
Pin 3	Data Pin 24
Pin 4	Data Pin 25
Pin 5	Data Pin 26
Pin 6	Data Pin 27
Pin 7	Data Pin 28
Pin 8	Data Pin 29
Pin 9	Data Pin 30
Pin 10	Data Pin 31
Pin 11	5V
Pin 12	GND

- If using a 1602a LCD or equivalent then wire the connections as follows:

LCD 1602A	Arduino
Pin 1	GND
Pin 2	5V
Pin 3	GND
Pin 4	Data Pin 22
Pin 5	GND
Pin 6	Data Pin 23
Pin 7	N/C
Pin 8	N/C
Pin 9	N/C
Pin 10	N/C
Pin 11	Data Pin 27
Pin 12	Data Pin 26
Pin 13	Data Pin 25
Pin 14	Data Pin 24
Pin 15	5V with 220ohm Resistor
Pin 16	GND

- If using a Noritake CU20025ECPB-U1J or equivalent then wire the connections as follows:

CU20025ECPB-U1J	Arduino
Pin 1	GND
Pin 2	5V
Pin 3	N/C
Pin 4	Data Pin 22
Pin 5	GND
Pin 6	Data Pin 23
Pin 7	Data Pin 31
Pin 8	Data Pin 30
Pin 9	Data Pin 29
Pin 10	Data Pin 28
Pin 11	Data Pin 27
Pin 12	Data Pin 26
Pin 13	Data Pin 25
Pin 14	Data Pin 24

Reference:

<https://media.digikey.com/pdf/Data%20Sheets/Noritake%20PDFs/CU20025ECPB-U1J.pdf>

- If using a DataVision DV-16236 or equivalent then wire the connections as follows:

DataVision	Arduino
Pin 1	GND
Pin 2	5V
Pin 3	N/C
Pin 4	Data Pin 22
Pin 5	GND
Pin 6	Data Pin 23
Pin 7	N/C
Pin 8	N/C
Pin 9	N/C
Pin 10	N/C
Pin 11	Data Pin 27
Pin 12	Data Pin 26
Pin 13	Data Pin 25
Pin 14	Data Pin 24

Reference: http://www.datavision.com.tw/en/lcm_01_1.php?P_Id=9

- If using a Noritake VFD GU140X16G-7003 or equivalent then wire the connections as follows:

VFD	Arduino
Pin 1	5V
Pin 2	Data Pin 3
Pin 3	GND
Pin 4	Data Pin 5
Pin 5	N/C
Pin 6	Data Pin 7

Reference: https://www.noritake-iron.com/site2017/images/Specs/VFD/MDG/GU140x16J-7000x/7003/gu140x16j-7003_e01-h8.pdf

- This assumes display is preset to Async Serial; if set otherwise then a different pinout and library will be required
- **IMPORTANT!** The sketch will hang on Initializing if you test it without the Noritake display attached
- **WARNING!** This display draws more power than my desktop PC USB could provide and exhibited strange behavior when all of the hardware was tested together. A 5W wall adapter provided sufficient power to prevent this condition

- Keypad Wiring (See Figure 2 for examples of tested keypads)
 - o If using the Bally 6x2 or 3x4 keypad then wire the connections as follows:

Bally	Arduino
Pin 1	Data Pin 39
Pin 2	Data Pin 38
Pin 3	Data Pin 41
Pin 4	Data Pin 40
Pin 5	Data Pin 43
Pin 6	Data Pin 42
Pin 7	Data Pin 45
Pin 8	N/C

- o If using the ACT 8x2 keypad then wire the connections as follows:

ACT	Arduino
Pin 1	Data Pin 38
Pin 2	Data Pin 39
Pin 3	Data Pin 40
Pin 4	Data Pin 41
Pin 5	Data Pin 42
Pin 6	Data Pin 43
Pin 7	Data Pin 44
Pin 8	Data Pin 45
Pin 9	N/C

- The Serial board needs to be wired as follows:

Serial	Arduino
VCC	5V
RXD	Data Pin 19
TXD	Data Pin 18
GND	GND

- If using an existing compatible Magnetic Stripe reader (See Figure 3) then wire the connections as follows:

Card Reader	Arduino
Pin 1 (5V)	5V
Pin 2 (Data)	Data Pin 20
Pin 3 (Clock)	Data Pin 21
Pin 4 (Unused)	N/A
Pin 5 (Insert)	Data Pin 8
Pin 6 (Inserted)	Data Pin 9
Pin 7 (Ground)	GND

- If using RFID then the reader needs to be wired into data pins 49-53 as follows:

RFID Reader	Arduino
SDA	Data Pin 53
SCK	Data Pin 52
MOSI	Data Pin 51
MISO	Data Pin 50
GND	GND
RST	Data Pin 49
3.3V	3.3V

You will need to power the Arduino board from the Accessory Outlet in the base of the slot machine using a USB power brick (5W or greater). A 6' USB cable is recommended to reach the top cabinet. It may be possible to get power from the top-cabinet but that is beyond the scope of this document

- Do not power the board from any built-in USB port on the game – it will not provide enough power

You will also need to run a CAT5 cable into the top box for the network or use a WiFi-RJ45 dongle – which will require separate power

Compatible WiFi dongle: <https://www.amazon.com//dp/B07QXNWRWJ>

INITIAL SETUP

- Assumes you have the Arduino IDE setup, the board and COM port settings are correct and all required libraries are installed, including
 - o IniFile*
 - o IeeFlipNoFrills, Noritake GU-7000 or LiquidCrystal
 - The sketch default configuration is a 20x2 LCD using the LiquidCrystal Library
 - The sketch defaults to 4 bit mode; there is an option for 8 bit mode
 - You will need to use the included modified libraries for the IeeFlipNoFrills or the Noritake displays
 - For the Noritake library there is another section of code to enable in the setup() function
 - The Noritake CU20025ECPB-U1J uses the LiquidCrystal library in 8 bit mode
 - o SPI*
 - o MFRC522* or MagStripe
 - You will need to use the modified MagStripe library included in the package
 - o Ethernet*
 - o SD*
 - o Keypad*
 - The sketch defaults to a Bally compatible 6x2 or 3x4 matrix
 - o TimeLib (<https://github.com/PaulStoffregen/Time>)
 - *Can be downloaded via the Arduino IDE Library Manager
- Format the MicroSD Card as FAT32
- Edit the included config.txt file with your settings and preferences; then copy it and the index.htm file to the SD card
 - o The scrolling text options have a max length of 255 characters
 - o Other text options have a limit of 30 characters (with exception of Casino Name – which should not exceed max width of your display)
 - o Other options are Boolean or Integers
 - o **IMPORTANT! Set the display dimensions to the correct size based on your display**
 - Some VFDs like the Noritake calculate the display size in pixels; Example: a GU140X16G-7003 is a 140x16 display; so the displayWidth should be 140 and the displayHeight should be 16
 - You will need to adjust the displayCols and displayRows to match the character dimensions of the display; Example: the GU140X16G-7003 emulates a 20x2 character display; so the displayCols should be 20 and the displayRows should be 2
 - Setting these incorrectly will cause display problems
- Insert the SD card into the slot on the Ethernet Shield
- Connect the board to your computer and load the ArduinoTITOPayerTrackingV2 or ArduinoTITOPayerTrackingV2MAG Sketch
- Modify the sketch as needed to enable support for your display and keypad; see inline comments for the relevant sections
- Open the Serial Monitor
- Download the sketch to the Arduino

- Wait for the app to finish initializing
- Disconnect the Arduino from your PC
- Mount the hardware into the slot machine's top box. Your specific player tracking bracket and options may vary – use your best judgement on how to mount the hardware
- For RFID I removed the existing card reader electronics (leaving the slot and frame) and mounted the RFID reader on top of the slot so it can read the cards when inserted
- Run the USB power wire down through the cabinet and route it to the accessory outlet
- Run the CAT5 Ethernet cable down through the cabinet and out to your network outlet
- Run the Serial Port cable down through the cabinet and connect it to the serial pigtail (if you are replacing a BETTORSlots TITO board; otherwise an adapter will need to be created) Running this cable may be a tight fit – you may need to remove the grommet in the cabinet pass-through or shave down the cable-end to make it fit
- Power on the machine and wait for it to fully start
- Insert your first card - the app will create a default player tracking file on the SD Card. Note the cardID in the Serial Monitor. This will be your first player card
- Remove the card from the reader and insert a different card. This will create a second card file which you can edit into the Admin card. Note the cardID in the Serial Monitor
- Remove the card from the reader and insert a different card. This will create a third card file which you can edit into the System Bonus card. Note the cardID in the Serial Monitor
- Unplug the Arduino; remove the SD Card and insert it into your PC/Laptop. Edit the player tracking card files accordingly. Details on the file formats are in the 'Arduino TITO and Player Tracking Project - File Formats.txt' file
- After editing the files replace the SD card in the Arduino
- Power on the machine and test
 - o Assumes you are replacing an existing TITO board or have already setup your machine per the instructions later in this document

CARD SERVER (OPTIONAL)

- Included in the package is a self-contained web server (APTS.EXE) designed to host and serve the card data to multiple slot machines. It can be run on any Windows 10 PC with the .NET Core 3.1 runtime (<https://dotnet.microsoft.com/download/dotnet-core/3.1>)
 - o If you have multiple machines (each with one of these boards) you can also designate one of the machines to host the card data and then set the other machines to be clients by setting their localStorage option to 0 and populating the serverIPAddress with the IP of the machine to act as host. That machine (or more specifically the board) will need to be powered for it to serve card data to the other machines
- Place the EXE in a folder on your PC. Cards will be created in this folder
- You will need to disable Windows Firewall or add a rule to allow connections to port 80
- The app must be run as an Administrator
- If you have configured your machines to use the card server and it is not running then player tracking will be unavailable
- In each config.txt file set the localStorage option to 0 and set serverIPAddress to the IP Address of the computer running the Card Server

ADMIN MENU

- The Admin Menu can be entered by inserting the Admin Card or pressing [CLR] on the keypad and then entering the Admin Pin followed by [ENT] on the keypad
 - o The Admin Pin defaults to 1234 and is adjustable in the config.txt file
- The display will change to 'ADMIN MENU'. In this mode you can perform the following actions from the keypad:
 - o Press 1 to Add Credits; after selecting this option use the keypad to enter the number of credits to add to the game
 - o Press 2 to Unmute the game
 - o Press 3 to Mute the game
 - o Press 4 to Unlock the game
 - o Press 5 to Lock the game
 - o Press 6 to Enable the Bill Validator
 - o Press 7 to Disable the Bill Validator
 - o Press 8 to Enable Change-to-Credits feature
 - o Press 9 to Disable Change-to-Credits feature
 - o Press 0 to Exit Admin Menu
- Remove the Admin Card or press '0' to return to the attract mode

PLAYER MENU

- The Player Menu can be entered by pressing [ENT] on the keypad while a Player Card is inserted.
- The display will change to 'PLAYER MENU'. In this mode you can perform the following actions from the keypad:
 - o Press 1 to Show Comp Balance
 - o Press 2 to Use Comp Credits
 - o Press 0 to Exit Player Menu
- Remove the Player Card or press '0' to return to the attract mode

PLAYER COMPS

Players can earn Comp Credits while playing the game at a rate specified by the 'compPercentage' setting in the config.txt file. It defaults to 0.01 comps per credit played – so if you play 100 credits you will earn 1 comp credit. Setting compPercentage to zero will disable the comp system.

Comp Credits can be redeemed at the machine by entering the Player Menu (instructions above) and using option 2. Only whole-credits can be redeemed.

REMOTE ACCESS

- The application includes a web interface for controlling various aspects of the game as well as the player tracking display
- It is compatible with the BETTORSlots IOS/Android apps (except tournament mode)

Arduino TITO and Player Tracking
Game Name: **GameKing** IP Address: **192.168.1.249**
Current player: **No Card Inserted**

Add Credits	1000
Sound On	Sound Off
Unlock Play	Lock Play
Enable Bill Validator	Disable Bill Validator
Change/Credits On	Change/Credits Off
Game Statistics	Player Statistics
Reset Handpay	Update Player Name

Update Display

Update Message

Update Ticket Info

Update Ticket Info

This device is compatible with the BETTORSlots Android/IOS apps for remote control.

Most functions are self-explanatory; The Change/Credits feature allows you to enable adding credits by pressing the Change (or Service) button. The number of credits added on each press is set in the config.txt file.

The Update Message option allows you to change the default scrolling text message; it resets to the value in the config.txt when the board is reset.

The Update Ticket Info option allows you to change the information printed on the cash-out ticket.

GAME MANAGER

Game Manager allows you to manage one or more game machines via a simple interface. It can be run on any Windows 10 PC with the .NET 5.0 runtime (<https://dotnet.microsoft.com/download/dotnet/5.0>)

Click 'Add Game' to add a new row to the grid. Only the IP Address is needed as the other fields will be populated by the game data. Double-click on the row's IP Address cell to edit it. Then press Refresh to query the game for data. You can delete a game by selecting the row in the grid and clicking 'Delete Game'. For the Game Controls – you can select one or more games to control/update by holding down the [CTRL] key while clicking on each row.

The screenshot shows the 'Arduino TITO and Player Tracking - Game Manager' application window. At the top is a table with columns: Game Name, IP Address, Arduino Clock, Is Online, Last Response, Credits, Total In, Total Won, Total Games, Games Won, Games Lost, Current Player, Games Played, Games Won, Games Lost, Total Won, Comps Earned, and Tournament Score. Below the table are three buttons: 'Add Game', 'Delete Game', and 'Save Config'. To the right are checkboxes for 'Apply to all games' and 'Auto Refresh', and a 'Refresh' button. The 'Game Controls' section includes buttons for 'Add Credits' (with a value of 100), 'Sound On', 'Sound Off', 'Unlock Play', 'Lock Play', 'Enable Bill Validator', 'Disable Bill Validator', 'Change/Credits On', 'Change/Credits Off', 'Reset Handpay', and 'Sync Date/Time'. Below this is the 'Tournament Mode' section with a 'Tournament Duration (in minutes)' dropdown set to 5, a 'Starting Credits' input set to 500, a 'Sync Arduino Clocks' button, a 'Tournament Clock' display showing 00:00, and 'End Tournament' and 'Start Tournament' buttons. On the right, the 'Update Display' section has a text input 'Enter your custom message here' and an 'Update Display' button. The 'Update Ticket Info' section has inputs for 'Casino Name/Location', 'Address Line 1', and 'Address Line 2', with an 'Update Ticket Info' button. A footer message reads: 'Reminder! Not all games support all features and options. Thank you.'

TOURNAMENT MODE

The Game Manager allows you to select two or more machines to play in a tournament mode. You can specify the duration and the starting credits. To begin tournament play each player must have a Player Tracking Card and it must be inserted in the machine they want to play. Each machine is checked to see if the player card is present and the correct number of credits are on the machine. If there are too many credits you will need to play it down or cash out the machine. The Game Manager will apply the correct number of starting credits when the game starts. There is a 30-second countdown and then the games will unlock and play can begin.

The player's score during the tournament will be visible in the Player Tracking Display. Players cannot add additional credits during tournament mode.

The tournament will run until the clock expires or the host can end the tournament early from the Game Manager. If a player removes their Player Card while the tournament is in progress or they run out of credits their game will end early and they will not be allowed back in until the tournament ends. The winner will be displayed in the Game Manager and on the winning machine's Player Tracking Display.

NOTES ABOUT TOURNAMENT MODE

- This is not SAS HOST Tournament mode; this is my own implementation which should work on any game compatible with my board
- Tournament mode depends on the games being in-sync with the host (Game Manager) clock. The Arduino hardware does not have a true real-time clock – so we have to do it in software. The clocks can drift slightly and network latency can slightly affect the times – but the Game Manager will attempt to update the clocks on each refresh
 - o Since there can be a delay in the HTTP response back from the board it is not uncommon for the displayed time to be a little off from one machine to the next – but the internal time on the board should be correct
- There is a 30-second countdown clock to the actual tournament start; this is to give both players and the machines enough time to respond to the Game Manager and set things up. Don't be surprised if some games respond faster than others – especially over WiFi. The games should start roughly at the same time (+- 1 second)
- It is not absolutely necessary for the games to be in perfect time-sync
- I have only tested this on two machines – and it worked, but your mileage may vary

NOTES ABOUT GAME MANAGER

- If a player card is inserted in a game you can edit the Player's Name by double-clicking on the name in the grid, changing it and then pressing [ENTER] or clicking away from the field
- My games did not always accept the Sync/Date Time command on the first try; not sure if this is an issue with the game or the data yet
- Build 20210222 includes the ability to update the config.txt file on your games without having to pull the SD cards from the board. You can use the 'Update SD Card' option to upload newer files from your PC. Be careful though! You can easily brick your device or cause it to return to default settings if you apply a bad config file
- Build 20210222 also includes the ability to remotely reboot the Arduino board
- If you hover over the game name you will see the board software version in the hint-text.
- This is definitely a BETA app; expect bugs

CONFIGURING YOUR IGT MACHINE (S2000/GameKing)

Before getting started with setting up your IGT machine for TITO please ensure your Bill Validator is working correctly and your Ticket Printer can print clear and legible tickets. You will also need a Keychip appropriate for your type of machine. These instructions assume familiarity with the Keychip process. Note – keychips and menu option locations vary from model-to-model. Please consult your IGT user manual or me if you have questions.

1. Clear any credits off your machine
2. Keychip your machine
3. Once in the Keychip Menu, ensure your Denomination, Devices, Limits and Game settings are as you want them
4. Setup the Comm Options as follows
 - a. IGT SAS Primary Channel = Channel 3
 - b. SAS Secondary = Off
 - c. Bally Miser = Off
 - d. Progressive Link = 7
 - e. WAMM 1.0 = Off
5. Setup the Validation/Redemption Options as follows
 - a. Validation = System Validation
 - b. Redemption = SAS Redemption
6. Setup the IGT SAS Options as follows
 - a. System Bonusing = SAS Legacy
7. Setup the SAS Channel (Primary Channel) Options as follows
 - a. Address = 1
 - b. Legacy Bonus = Enabled (X)
 - c. Validation = Enabled (X)
8. Setup the Machine Terminal Options as follows
 - a. Voucher Limit Follows = Credit Limit
9. Save all options and make any other changes you wish before pressing Return to Game to exit the keychip menu

Test the game by inserting money and then pressing Cash Out to generate a ticket. The ticket serial number should match the number of credits you inserted. Insert the ticket into the machine, it should accept it for the same number of credits.

CONFIGURING YOUR IGT MACHINE (AVP)

Please see this video for how to configure your AVP Game:

<https://www.youtube.com/watch?v=JKIyeFQPltA>

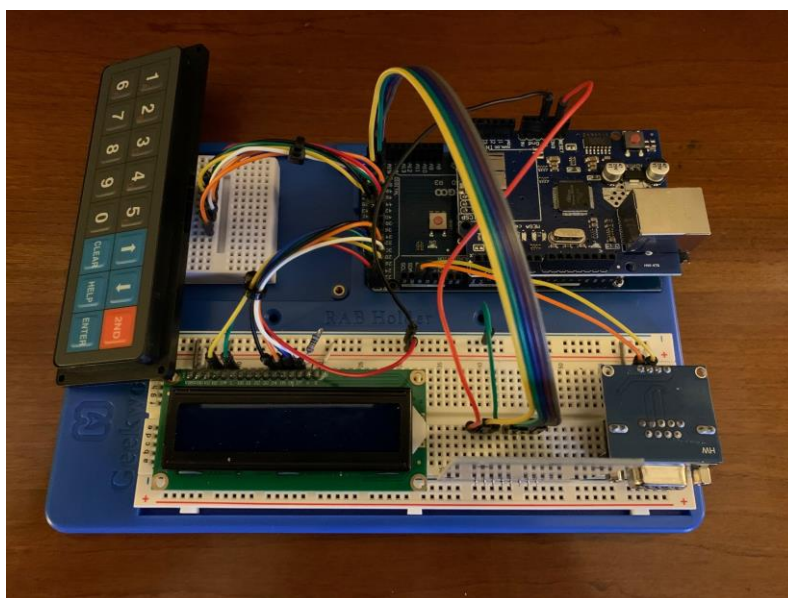
NOTES

- This project uses specific hardware and libraries. It may be possible to switch out the Ethernet Shield for another. I DO NOT RECOMMEND using a WiFi Shield – had nothing but problems during testing
- If you do not have a display or card slot (or only wish to use this as a TITO solution) you can set the option 'onlyTITO' to 1 in the config.txt
- Not all machines support the Audio Mute or the Remote Handpay Resets – in those cases the commands will be ignored by the machine
- A user reported erratic behavior on an IGT AVP game; it was traced to interference on the serial cable; to fix it they used a shielded DB9 cable and grounded the cable to the game chassis
- I was only able to successfully test the Bally Magnetic Card reader pictured in Figure 3. The Neuron card reader should have the same pinouts however the one I have may be defective as the switch pins were floating high randomly. Your mileage may vary

DISCLAIMER

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

PHOTOS



Mockup using 1602LCD, 8x2 Keypad and RFID reader

FIGURE 1: IGT Serial Cable



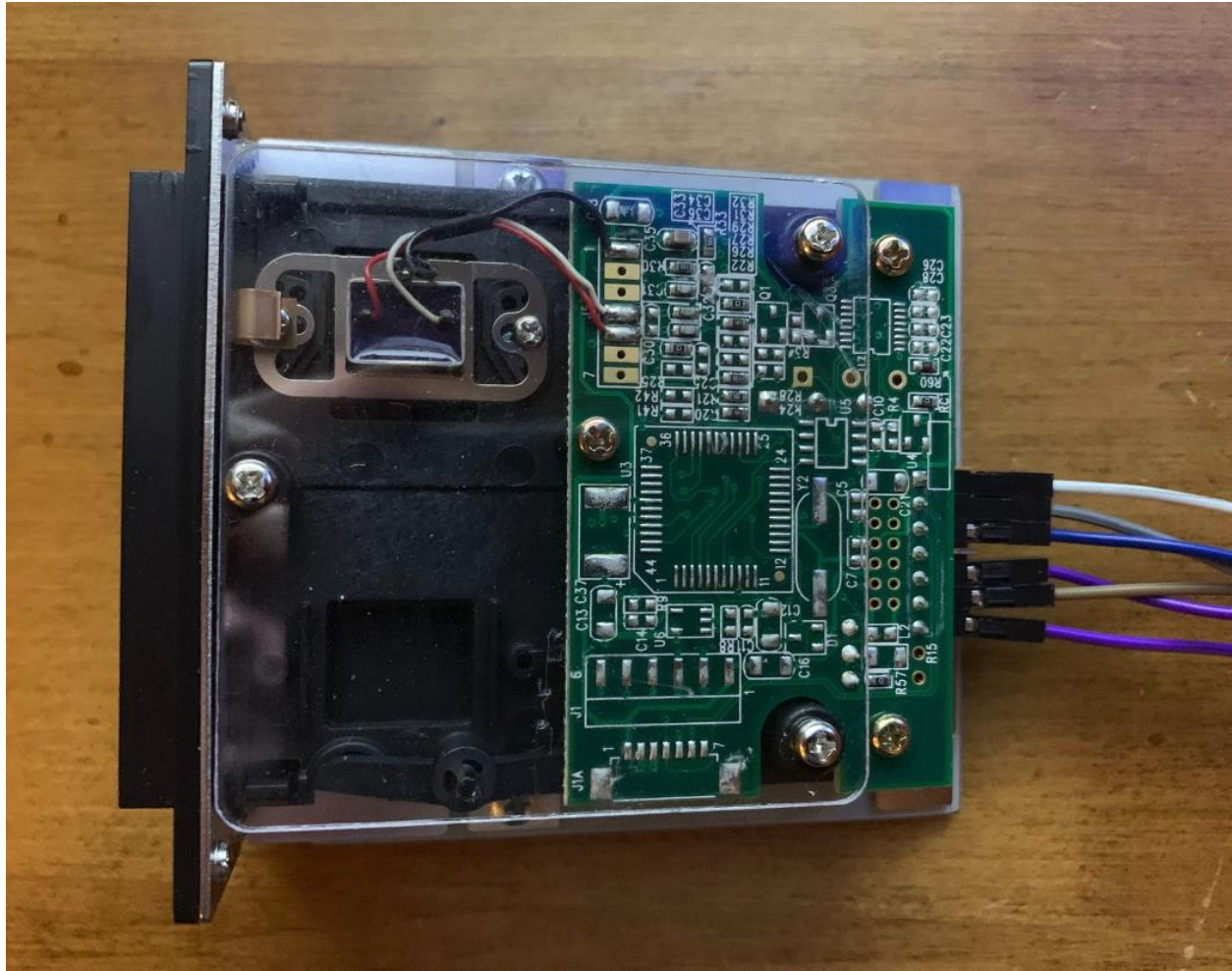
IGT S2000/GameKing Serial Pinout

DB9 Male	Signal direction	S2000 J82
2 Receive Data	←	1 Transmit Data
3 Transmit Data	→	2 Receive Data
5 Signal Ground	---	5 Signal Ground

FIGURE 2: Compatible/Tested Keypads



FIGURE 3: Compatible/Tested Card Readers



UNIFORM INDUSTRIAL CORP; MSR240-02TMRNWWBR

Pin 1 in this orientation is at the top (white)