PACHINKO GAMES & COUNTER

www.pachinkoplanet.com

Carl Wegman
Sugar Grove, Illinois, USA
carlwegman@aol.com

Table of Contents

What is it?

The Game is an “add on device” for a pachinko machine.
The Game keeps score and saves the top 5 high scores for two different Games: **Timer** Game and **Counter** Game.

How does it work?

After installation, turn on the power switch and the Game will act like a pachinko game counter, recording pachinko balls SHOT and pachinko balls WON, the balance (SCORE) is computed and displayed, the elapsed time is displayed.

At any time, follow the prompts on the LCD display to play either of the two Games or to change the default values.

The pachinko machine will play exactly the same as it did.

There are two sensors that detect each SHOT pachinko ball and each WIN pachinko ball. The WIN sensor can be replaced with a microswitch (included); if that is easier to install on your pachinko machine. There are legacy (GAKU) connectors included if you have an existing GAKU counter installed; the GAKU sensors can be used on this Game with no wiring modifications required. The sensors’ placement / location may need to be adjusted.

Why play this Game?

This Game will allow up to 4 players to compete in two different Games.
The top 5 high scores are recorded and saved in the Game (no batteries required).
This game allows players to compete or keep track of scores based upon time or pachinko balls shot.

Can the Game be customized to match my décor or my machine?

The wood can be painted or stained, by you, to match your décor or your machine. The clear plastic faceplate of the game can be removed and the paper faceplate template can be replaced to suit your needs.
The Game’s *programming* can be modified and updated; the Game is installed on an Arduino MEGA platform.

Will the Game work on modern pachinko machines?

The Game will work on almost **ANY** kind of pachinko machine: Vintage, Early Modern, Hanemono, Modern, Cyclic…
As long as the sensors can be suitably mounted, the Game will work.

Revision 11 MEGA 2013 – January – 2013 – Game Descriptions

Game Menu (Press *)

Timer Game

- 1 to 4 players selectable
  - Can use default player name
  - Can change player name (not stored as a default name)
- Timer Game Length from 30 seconds to 10 minutes (600 seconds)
  - Can use a default game time length
  - Can select game time length (not stored as a default value)
- If game time length is the same as the default value, the top 5 high scores will be saved in EEPROM memory and high scores will remain in memory (with power off) until the stored value for the game time length is changed OR until ALL stored values are restored OR until a higher score is recorded.
- Game starts after first shot is sensed.
- Time will be counted down after first shot.
- After all players have taken their turn, scores will be displayed and players will be ranked.

Counter Game

- 1 to 4 players selectable
  - Can use default player name
  - Can change player name (not stored as a default name)
- Counter Game from 10 to 1,000 shots
  - Can use a default number of shots
  - Can select the number of shots (not stored as a default value)
- If the number of shots is the same as the default value, the top 5 high scores will be saved in EEPROM memory and will remain in memory (with power off) until the stored value for the number of shots is changed OR until ALL stored values are restored OR until a higher score is recorded.
- Game starts after first shot is sensed.
- Time will be counted up after first shot.
- After all players have taken their turn, scores will be displayed and players will be ranked.

Default Menu (Press D) "Changeable" Default Values:

Location	Value Name	Digits	(Stored as Shown) (Stored in ASCII) Stored Value
1	Player 1 Name	12	Player 1
2	Player 2 Name	12	Player 2
3	Player 3 Name	12	Player 3
4	Player 4 Name	12	Player 4
5	Timer Game Length (30 to 600 Seconds)	3	60
6	Contrast (if applicable - 0 to 62)	2	20
7	Shot Counter Game Shots (10 to 1,000)	4	100
8	* Shot LED Enabled	n/a	1 = Enabled
11	Shot LED Duration (in milliseconds)	4	250
9	* Win LED Enabled	n/a	1 = Enabled
12	Win LED Duration (in milliseconds)	4	1000
10	* Bell Relay Enabled (via relay)	n/a	1 = Enabled
13	Shot LED Duration (in milliseconds)	4	110
14	Reward Balls per Win (can not be 0)	2	15
35	Delay Multiplier Enabled	n/a	1 = Enabled
36	Delay Multiplier (times)	1	5
47	Bell ON / OFF Button Enabled	n/a	1 = Enabled
48	Bell Button LED Enabled	n/a	1 = Enabled
49	Win Microswitch Enabled	n/a	0 = disabled
50	* Servo Relay Enabled	n/a	0 = disabled
51	Servo Duration (milliseconds)	4	250
n/a	Restore ALL Stored Values	n/a	n/a

"Changeable" Default NOTES:

High scores will only be recorded if:

- The timer game length is the same as the default value.
- OR
- The number of shots in the shot counter game is the same as the default value.

If the default value for the timer game length is changed; the timer game high scores will be RESET to the default values.

If the default value for the number of shots in the shot counter game is changed; the counter game high scores will be RESET to the default values.

The following default values require the game power to be restarted to become effective:

- * Shot LED Enabled
- * Win LED Enabled
- * Bell Relay Enabled
- Bell ON / OFF Button Enabled
- Bell ON / OFF Indicator LED Enabled
- * Servo Relay Enabled
- Servo Button Enabled

* If any of the above defaults are "Enabled", an additional default will be available to adjust the time length of the default (in milliseconds; 1000 ms = 1 second). This time length can not be set to 0.

The Delay Multiplier is used to "pause" the display for a number of display screens e.g. Game Score, High Score, etc. The amount of delay can be adjusted up or down but can not be set to 0.

The Bell ON / OFF Indicator LED indicates if the Bell will "ring". If the bell is "ON" (and the indicator LED is enabled); the Bell Indicator LED will also be "ON" and remain on. If the bell is "OFF"; the Bell Indicator LED will also be off.

If the Win Microswitch is Enabled; the Win Microswitch (low force, wired normally closed) is used to sense Wins INSTEAD of the Win Sensor.

If the Servo Relay is Enabled; the servo relay is energized for the default servo duration prior to each Players' turn in a game. The servo button will also energize the servo relay for the default servo duration.

When editing Names; press the "A=Accept" key when the name appears as desired. Don't press the "#=Next Name Letter" key.

When entering numbers for game length, shot count, etc. there are a limited number of digits; if the limit is four digits as soon as you key the fourth digit – your number entry is validated and if valid, number entry is completed.

Additional NOTES:

The Sensors are wired to be compatible with the GAKU Counter. If you have a "working" GAKU Counter; the sensors can be left in place and plugged into the corresponding plugs on this Game. The sensor location / placement may need to be adjusted.

Important notes about the legacy (GAKU) sensor connections

- Exercise care with metal objects near the connectors, 5V is present and shorting out the power will cause electric shock and will damage the game electronics.
- Do not have legacy sensors connected at the same time as the regular game sensors.
- Turn off the game power when connecting or disconnecting any of the connectors.
- The regular game sensors are on a polarized connector with a clip. The connector will easily insert into the receptacle. The connector needs to be "unlatched" to remove the connector. However, the legacy (GAKU) sensors are not polarized. Exercise care to plug in the connectors according to the colors on the game printed circuit board. Refer to the game printed circuit board and the attached illustrations.

The relays are high current relays with a capacity of: AC250V 10A; DC30V 10A. The bell relay is suitable for use with a standard door bell (8V – 24V AC). The servo relay is suitable for use with a Power Flash (requires wiring to the Power Flash Microswitch aka wiring a "cheat button"; may require diodes to suppress line noise.

The front panel can be removed and the paper template can be replaced or customized to match your Machine. Use the plastic face plate as a template.

The processor is an Arduino Mega 2560 R3 and can be customized upon request.

The contrast of the display is adjusted via a small, blue set screw on the back of the display or through the defaults menu (if available). Once set to your voltage, it will not need frequent adjustment. IMPORTANT: Use caution to not touch any metal to the printed circuit boards.

High Scores and Default Values are stored in EEPROM memory and remain after power is turned off, no batteries are required.

During Menu Selections (any time the keypad is being used), the front panel push buttons are not being polled for activity. This means the bell can not be turned on and off, shots and wins will not be sensed, the servo can not be

energized, etc. The exception is the Reset button which resets the Arduino processor and the display.

Scores ranges are limited: from -999 to 9999.

When on the Game Menu, pressing the # key on the keypad will return to the Initial Display Screen.

The timer displayed on the screen has a limit of 99 minutes and 98 seconds and will then reset to 0. There are other instances during the program where the time displayed counts backwards OR resets to 0. There is an upper limit to the timer; it will overflow (go back to zero), after approximately 50 days if the power is left on continuously. Try a reset or power off; if the timer is not behaving as expected. The timer is not as accurate as a real clock; however it will be consistent for each player.

If the bell relay is enabled, the bell will be on when the game is powered up; the bell indicator LED will also be on.

If the game has false readings some trouble shooting tips to try:

- If you have a modern pachinko machine with a shooting handle; look for an unattached green wire. **Make sure this is a ground wire.** Attach it to the negative lead of the game shield via the screw type terminal.
- If you have a vintage machine and you are using a separate power supply for the lights; Connect the ground of all **DC** power supplies. Exercise **extreme care** to assure correct polarity and **DO NOT** hook up AC power to the game's ground. **DO NOT** hook up a positive lead to the game's ground.
- If your pachinko machine has solenoids, solder appropriate diodes across the solenoid leads. Ensure proper polarity of the diodes.

Default Values (ALL)

These values are installed during first initialization of the program. Subsequently, default values are stored and read from non-volatile EEPROM memory. No battery is required. These default values (or changed, stored values) remain in memory after power is turned off.

Default Value 1 - Player 1 Name = "Player 1 " in ASCII
Default Value 2 - Player 2 Name = "Player 2 " in ASCII
Default Value 3 - Player 3 Name = "Player 3 " in ASCII
Default Value 4 - Player 4 Name = "Player 4 " in ASCII
Default Value 5 - Default time (seconds) for Game A Timer Game three digits = "60 " in ASCII
Default Value 6 - Default Contrast setting 2 digits from 0 to 62 on DigiOle display = "20" in ASCII
Default Value 7 - Default number of shots for Game B Shot Counter Game four digits = "100 " in ASCII
Default Value 8 - Shot LED Enabled = 1 = Enabled
Default Value 9 - Win LED Enabled = 1 = Enabled
Default Value 10 - Bell Relay Enabled = 1 = Enabled
Default Value 11 - Shot LED Duration (milliseconds) four digits = "250 " in ASCII
Default Value 12 - Win LED Duration (milliseconds) four digits = "1000" in ASCII
Default Value 13 - Bell Duration (milliseconds) four digits = "1100" in ASCII
Default Value 14 - Balls per Win two digits = "15" in ASCII
Default Value 15 & 30 - Player Name High Score 1 = "Carl W " in ASCII
Default Value 16 & 31 - Player Name High Score 2 = "Kyle W " in ASCII
Default Value 17 & 32 - Player Name High Score 3 = "Daniel W " in ASCII
Default Value 18 & 33 - Player Name High Score 4 = "Russell W " in ASCII
Default Value 19 & 34 - Player Name High Score 5 = "Mary W " in ASCII
Default Value 20 & 25 - Default High Score 1 - 4 Digits = "150 " in ASCII
Default Value 21 & 26 - Default High Score 2 - 4 Digits = "140 " in ASCII
Default Value 22 & 27 - Default High Score 3 - 4 Digits = "130 " in ASCII
Default Value 23 & 28 - Default High Score 4 - 4 Digits = "120 " in ASCII
Default Value 24 & 29 - Default High Score 5 - 4 Digits = "110 " in ASCII
Default Value 35 - Delay Multiplier enabled = 1 = Enabled
Default Value 36 - Delay Multiplier (times) - one digit = "5" in ASCII
Default Value 47 - Bell On Off Momentary Contact Button Enabled = 1 = Enabled
Default Value 48 - Bell Button LED Enabled = 1 = Enabled
Default Value 49 - Win Microswitch Enabled = 0 = disabled
Default Value 50 - Servo Relay Enabled = 0 = disabled
Default Value 51 - Servo Duration (milliseconds) four digits = "500 " in ASCII

4 x 20 Display Samples

```
Press * for Games
D for Defaults
SHOT  WON  TIME SCORE
0     0    0:00  0000
```

```
* for Games
Press Game Letter!
A. Timer Game
B. Shot Counter Game
C. Replay Last Game ( if available )
```

```
A. Timer Game
Press 1 to 4 Players
```

```
Use Default Game
Time Length?
60  seconds
#=Accept C=Change
```

```
Enter Game Length
In Seconds 30 to 600
#=Accept C=Clear
```

```
Use Default Player
Name = Player # Name
#=Accept C=Change
```

```
12 letters
A=Accept C=Clear
B=Back D=Next
#=Next Name Letter
```

```
Player # Name's Turn
Shot Starts Game
SHOT  WON  TIME SCORE
0     0    0:00  0000
```

```
TIMER GAME
   SCORE     0000
    Player Name
    000 Seconds
```

4 x 20 Display Samples (Continued)

B. Shot Counter Game

```
Press 1 to 4 Players

Use Default Number
of Shots?
100  Shots
#=Accept C=Change

Enter # of Shots
 10 to 1000
#=Accept C=Clear

Use Default Player
Name = Player # Name
#=Accept C=Change

12 letters
A=Accept C=Clear
B=Back D=Next
#=Next Name Letter

Player # Name's Turn
Shot Starts Game
SHOT  WON  TIME SCORE
0     0    0:00  0000

COUNTER GAME
   SCORE     0000
    Player Name
    0000 Shots
```

4 x 20 Display Samples (Continued)

```
D for Defaults
D = Defaults
C = Previous Score
A = High Scores
# = Return

D = Defaults
Player 1 Name =
Player 1
# = OK; C = CHANGE

Player 2 Name =
Player 2
# = OK; C = CHANGE

Player 3 Name =
Player 3
# = OK; C = CHANGE

Player 4 Name =
Player 4
# = OK; C = CHANGE

Game Time Length =
60  seconds
# = OK; C = CHANGE

# of Shots =
100  shots
# = OK; C = CHANGE

Shot LED Enabled =
Enabled
# = OK; C = CHANGE

Changed Default ##
Power Off Required!

Shot LED Duration =
250  milliseconds
# = OK; C = CHANGE

Shot LED Duration
<= 4 Digits in ms.
#=Accept C=Clear
```

4 x 20 Display Samples (Continued)

```
Win LED Enabled =
Enabled
# = OK; C = CHANGE

Changed Default ##
Power Off Required!

Win LED Duration =
1000 milliseconds
# = OK; C = CHANGE

Win LED Duration
<= 4 Digits in ms.
#=Accept C=Clear

Bell Relay =
Enabled
# = OK; C = CHANGE

Changed Default ##
Power Off Required!

Bell Duration =
1100 milliseconds
# = OK; C = CHANGE

Bell Duration
<= 4 Digits in ms.
#=Accept C=Clear

Balls per Win =
15 pachinko balls
# = OK; C = CHANGE

Balls per Jackpot
<= 2 Digits
#=Accept C=Clear

Delay Multiplier =
Enabled
# = OK; C = CHANGE

Changed Default ##
Power Off Required!
```

4 x 20 Display Samples (Continued)

```
Delay Multiplier =
5 times
# = OK; C = CHANGE

Delay Multiplier
<= 2 Digits
#=Accept C=Clear

Bell Button =
Enabled
# = OK; C = CHANGE

Changed Default ##
Power Off Required!

Bell Button LED =
Enabled
# = OK; C = CHANGE

Changed Default ##
Power Off Required!

Win Microswitch =
disabled
# = OK; C = CHANGE

Changed Default ##
Power Off Required!

Servo Relay =
disabled
# = OK; C = CHANGE

Changed Default ##
Power Off Required!

Servo Duration =
500  milliseconds
# = OK; C = CHANGE

Servo Duration
<= 4 Digits in ms.
#=Accept C=Clear
```

4 x 20 Display Samples (Continued)

```
Use Initial Default?
Will Erase ALL
High Scores & Names
# = NO; 1 = YES

Restored Initial
Settings to Default
Power Off Required!

PACHINKO GAME
   Version 3 Mega
               2013

C = Previous Score
1 Player Name 0000
2 Player Name 0000
3 Player Name 0000
4 Player Name 0000

A = High Scores

    Timer Game
High Score # 0000
Player Name

   Counter Game
High Score # 0000
Player Name
```

Sensor Installation

From the GAKU Counter Manual:

> *Installation of sensors: The "Shot" sensor goes at the shooting hummer, and the "Win" sensor at the dispensing motor. Use adhesive tapes to place sensors. You need to make adjustments until you find the right positions. Once the counter starts work accurately, use the "Hot Glue" to secure the sensors. Do not use permanent adhesives such as Super Glue or epoxy glue -- You might need to adjust again later.*

Spacers may be required behind the sensor to achieve satisfactory results.

Pachinko balls must pass directly under the center of the sensor, spacers may be required to “channel” the pachinko balls under the center of the sensor.

Start with a ½” space between the sensor and the bottom of the pachinko ball. Adjust until the pachinko ball is sensed consistently.

Small fasteners can be used to permanently mount the sensors; ideally leave room for further adjustment.

The sensors are sensitive to reflections; smooth plastic or metal can be the cause of false readings.

Connect the ground of all **DC** power supplies. Exercise **extreme care** to assure correct polarity and **DO NOT** hook up AC power to the game’s ground.
DO NOT hook up a positive lead to the game’s ground.

The Win sensor can be replaced by the Win Microswitch. Vintage machines are a good candidate for the Win Microswitch.

Important notes about the legacy (GAKU) sensor connections

- Exercise care with metal objects near the connectors, 5V is present and shorting out the power will cause electric shock and will damage the game electronics.
- Do not have legacy sensors connected at the same time as the regular game sensors.
- Turn off the game power when connecting or disconnecting any of the connectors.
- The regular game sensors are on a polarized connector with a clip. The connector will easily insert into the receptacle. The connector needs to be “unlatched” to remove the connector. However, the legacy (GAKU) sensors are not polarized. Exercise care to plug in the connectors according to the colors on the game printed circuit board. Refer to the game printed circuit board and the attached illustrations.

Housing Installation

Suggestions for removing the paper template:

- Remove the game from the housing
- Gently pull LED's from LED Holders
- Loosen push button washers and nuts and remove push buttons
- Remove plastic caps on 10 face plate screws
- Remove face plate machine screws
- Power switch is difficult to remove; see illustrations to view the four plastic tabs that need to be simultaneously pushed in to allow switch to be removed. Additional clear face plates are available upon request.
- Use the plastic face plate as a template for your paper insert; cut portions of paper that need to be removed with an *Exacto* knife and cutting mat.

If the game is enclosed; allow for ventilation.

The housing can be installed on top of the machine, in front of the Machine or in other ways to suit your Machine.

Remove the game from the housing and finish the wood to suit your needs. The veneer is white oak. A clear finish (e.g. Minwax® Polycrylic® Protective Finish) is a reasonable match to a lot of vintage machines. The wood can be painted, stained or laminated.

Temporary wood screws can be placed in the back surface to allow painting and staining of all surfaces.

The face plate/game is secured to the housing with 2 fasteners. **DO NOT** over tighten the two fasteners securing the face plate to the housing.

There are 2 mounting holes pre drilled for fastening the housing to the top of your pachinko machine. The housing can also be mounted in other ways to suit your needs.

DO NOT over tighten two screws / washers!

Arduino and Power Supply Notes:

The Arduino Mega can be powered via the USB connection (for updates only) or with an external power supply. The power source is selected automatically. An external power source is required to reliably utilize the display screen and the relay board.
External (non-USB) power can come either from an AC-to-DC adapter or a different power supply. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack or through the wire terminals on the corner of the shield. It is very important to connect positive and negative correctly. Incorrect polarity will damage the Arduino.
The board and shield can operate on an external supply of 7 to 12 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.
Changes to the power supply voltage or to the connected load may cause the display contrast to require adjustment. There is a small adjustment screw on the back pack of the display. Use a small, plastic screwdriver to adjust the contrast. The display should not require frequent contrast adjustments.

Arduino Shield Notes:

The shield attached to your game utilizes polarized, latched connectors wherever possible. These connectors promote simple positive connections between the components. The shield should not have to be disassembled to change the artwork behind the face plate.
If you do have a reason to remove the shield or the cables attached to the shield, PLEASE BE CAREFUL and remember:

- Never work on the device with power applied.
- Always expect a fire and monitor for excessive heat buildup while testing operation. Be ready to disconnect power and have a fire extinguisher ready.
- The display and the LEDs use the same size connector and are near each other, exercise care to assure that the correct connectors are mated. Similarly, the WIN Microswitch and the reset push button should be connected properly. See the illustrations for more details.
- Relays are used in place of momentary (switch) push buttons; the relay board may be connected to a NON-polarized connector. It is important to note the orientation of the connector and ensure that it is re-connected properly.
- It is normal for LIVE electric voltage and electric current to be present on the shield while power is applied. Metal objects (rings, bracelets, screwdrivers, tools, other) can cause an electrical shock or damage the electronics.
- Accept that while modifying electronics, you may damage them.

There may be unsoldered, single pins between the shield and the Arduino. These pins are for shield mounting stability. These pins can be removed for future expansion.

The position and names of a correctly assembled shield with an attached relay board (NOTE: relay boards vary) are shown in the illustrations.

Arduino Software Disclaimer:

Hardware Specifications and Notes:

The Arduino Mega 2560 is a microcontroller board based on the ATmega2560. It has 54 digital input/output pins (of which 14 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The Mega is compatible with most shields designed for the Arduino Duemilanove or Diecimila.
The Mega 2560 is an update to the Arduino Mega, which it replaces.
Revision 11 of the board has the following new features:
1.0 pinout: added SDA and SCL pins that are near to the AREF pin and two other new pins placed near to the RESET pin, the IOREF that allow the shields to adapt to the voltage provided from the board. In future, shields will be compatible both with the board that use the AVR, which operate with 5V and with the Arduino Due that operate with 3.3V. The second one is a not connected pin, that is reserved for future purposes.
Stronger RESET circuit.
Atmega 16U2 replace the 8U2.

Summary

Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Digital I/O Pins	54 (of which 15 provide PWM output) (*38 Available*)
Analog Input Pins	16 (*8 Available*)
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz

A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.

IOREF. This pin on the Arduino board provides the voltage reference with which the microcontroller operates. A properly configured shield can read the IOREF pin voltage and select the appropriate power source or enable voltage translators on the outputs for working with the 5V or 3.3V.

Memory

The ATmega2560 has 256 KB of flash memory for storing code (of which 8 KB is used for the bootloader), 8 KB of SRAM and 4 KB of EEPROM (which can be read and written with the EEPROM library).

Input and Output

Each of the 54 digital pins on the Mega can be used as an input or output, using pinMode(), digitalWrite(), and digitalRead() functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:
Serial: 0 (RX) and 1 (TX); Serial 1 (*Used by Game*): 19 (RX) and 18 (TX); Serial 2: 17 (RX) and 16 (TX); Serial 3: 15 (RX) and 14 (TX). Used to receive (RX) and transmit (TX) TTL serial data. Pins 0 and 1 are also connected to the corresponding pins of the ATmega16U2 USB-to-TTL Serial chip.

Hardware Specifications and Notes: (Continued)

External Interrupts: 2 (interrupt 0 – *Used by Game*), 3 (interrupt 1 - *Used by Game*), 18 (interrupt 5), 19 (interrupt 4), 20 (interrupt 3), and 21 (interrupt 2). These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value.
PWM: 2 to 13 and 44 to 46. Provide 8-bit PWM output with the analogWrite() function.
SPI: 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS). These pins support SPI communication using the SPI library. The SPI pins are also broken out on the ICSP header.
LED: 13. There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.
TWI: 20 (SDA) and 21 (SCL). Support TWI communication using the Wire library.
The Mega2560 has 16 analog inputs, each of which provide10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin and analogReference() function.
There are a couple of other pins on the board:
AREF. Reference voltage for the analog inputs. Used with analogReference().
Reset. Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

Programming

The Arduino Mega can be programmed with the Arduino software (download). For details, see the reference and tutorials.
The ATmega2560 on the Arduino Mega comes preburned with a bootloader that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol (reference, C header files).
You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header; see these instructions for details.

USB Overcurrent Protection

The Arduino Mega2560 has a resettable polyfuse that protects your computer's USB ports from shorts and overcurrent. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.

Physical Characteristics and Shield Compatibility

The maximum length and width of the Mega2560 PCB are 4 and 2.1 inches respectively, with the USB connector and power jack extending beyond the former dimension.
Four screw holes allow the board to be attached to a surface or case. Nylon machine screws, nuts and spacers were used for attaching the game to the wood face plate. These fasteners were used to minimize electrical short circuits.
The game shield does pass through all unused Arduino pins; so that an additional "proto" shield can be easily stacked on top of the game shield. Additional hardware, software and knowledge are required to extend the game shield's functionality. Proto shield vary; please ensure your project will fit upon the game shield. See illustrations.
A reset button is present on the game shield that resets the Arduino and the display screen. The on board reset button functions exactly the same as the reset button on the front of the face plate. The jumper on the shield causes the display to be reset any time the Arduino is reset.
There may be unsoldered, single pins between the shield and the Arduino. These pins are for shield mounting stability. These pins can be removed for future expansion.
Brass machine screws were used for the display. These fasteners were used to minimize galvanic corrosion.
The LEDs can be gently pulled out of the metal LED holders. Push the plastic piece behind the LED into the LED holder to re-install.

PACHINKO GAME Version 3 Mega 2013

Function Listing

(NOTE: Unless you have access to the source code for the game program and interest in the Arduino IDE, this page and the next page are not worthy of reading)

General Notes and Comments
Libraries & Variables
Void Function setup
Void Function loop - MAIN PROGRAM
Void Function CheckSensors
Void Function UpdateDisplay
Void Function WriteDefaults
char Function GetKeypadInput
Void Function ChangeDefaults
Void Function GameMenu
Void Function ChangeName
Void Function GetAGameTimeLength
Void Function GetBGameShotNumber
Void Function GetPlayers
Void Function TimerGame
Void Function CounterGame
Void Function ValidateDigits
int Function ReadValue
Void Function RecordScore
Void Function DisplayScreen
Void Function ShowOptions
Void ShotwasSensed – Interrupt 0
Void WinwasSensed – Interrupt 1

Arduino:

Software Version – 1.0.4
Memory Usage - 28,888 of 258,048
TOTAL Lines of Code ≈ 2933
Libraries Included in Game Code:

- EEPROM
- Bounce (Added to IDE)
- Keypad (Added to IDE)

Arduino MEGA-R3 notes:

Arduino Mega Pin Map - Shield 11

Pin	Description
2	Shot Sensor (Interrupt 0)
3	Win Sensor (Interrupt 1)
A0	Keypad
A1	Keypad
A2	Keypad
A3	Keypad
A4	Keypad
A5	Keypad
A6	Keypad
A7	Keypad
18	Serial Display RX (Serial 1)
19	Serial Display TX (Serial 1)
32 n/a	Serial Display RX (Software Serial) (requires code)
33 n/a	Serial Display TX (Software Serial) (requires code)
34	Bell On Off Button
35	Win Microswitch
36	Shot LED
37	Win LED
38	Bell Button Switch LED
39	Bell Relay
40	Servo Relay
41	Servo Relay Button
42 n/a	Microswitch 2 (requires code)
43 n/a	Microswitch 3 (requires code)

AVAILABLE ARDUINO PINS
(Pins Passed Through Shield)
SPI (ICSP) Bus / Header, IOREF, RESET, 3V3, 5V (Qty=3), GROUND (Qty=4), V IN, A8 through A15 (analog), SCL, SDI, AREF,
0 & 1 (digital), 4 through 17 (digital),
20 through 31 (digital), 44 through 53 (digital)

Available Pin Summary: 8 analog, 38 digital

Game Customization Questionnaire

1. Q. Do you have a GAKU counter installed on your pachinko machine?
A. Yes, No, Don't know

A. _____

2. Q. Will the game be installed on a "Power Flash" style pachinko machine?
A. Yes, No, Don't know

A. _____

3. Q. Is your pachinko machine "cyclic"?
A. Yes, No, Don't know

A. _____

4. Do you want an additional electric bell on your pachinko machine?
A. Yes, No, Don't know

A. _____

5. Q. Do you want additional space at the top of the game for a light fixture?
A. Yes, No, Don't know

A. _____

6a. Q. Do you want the default player names hard coded into the game?
A. Yes, No, Don't know

A. _____

If answer to question 6a is "Yes":
6b. Q. Names desired? (Up to 12 characters, case sensitive, spaces and punctuation marks are acceptable)

A. Player 1 __ __ __ __ __ __ __ __ __ __ __ __

A. Player 2 __ __ __ __ __ __ __ __ __ __ __ __

A. Player 3 __ __ __ __ __ __ __ __ __ __ __ __

A. Player 4 __ __ __ __ __ __ __ __ __ __ __ __

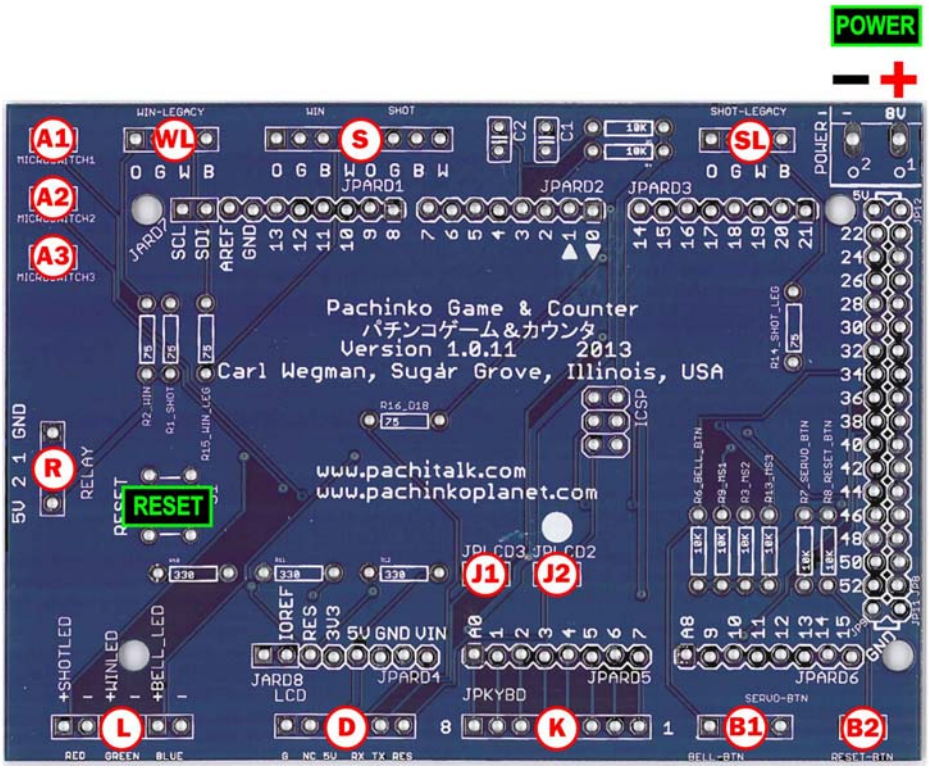
A. Player 5 __ __ __ __ __ __ __ __ __ __ __ __ (High Score only)

7. Q. Do you want a new 24V transformer to provide power for your pachinko machine AND the game?
A. Yes, No, Don't know

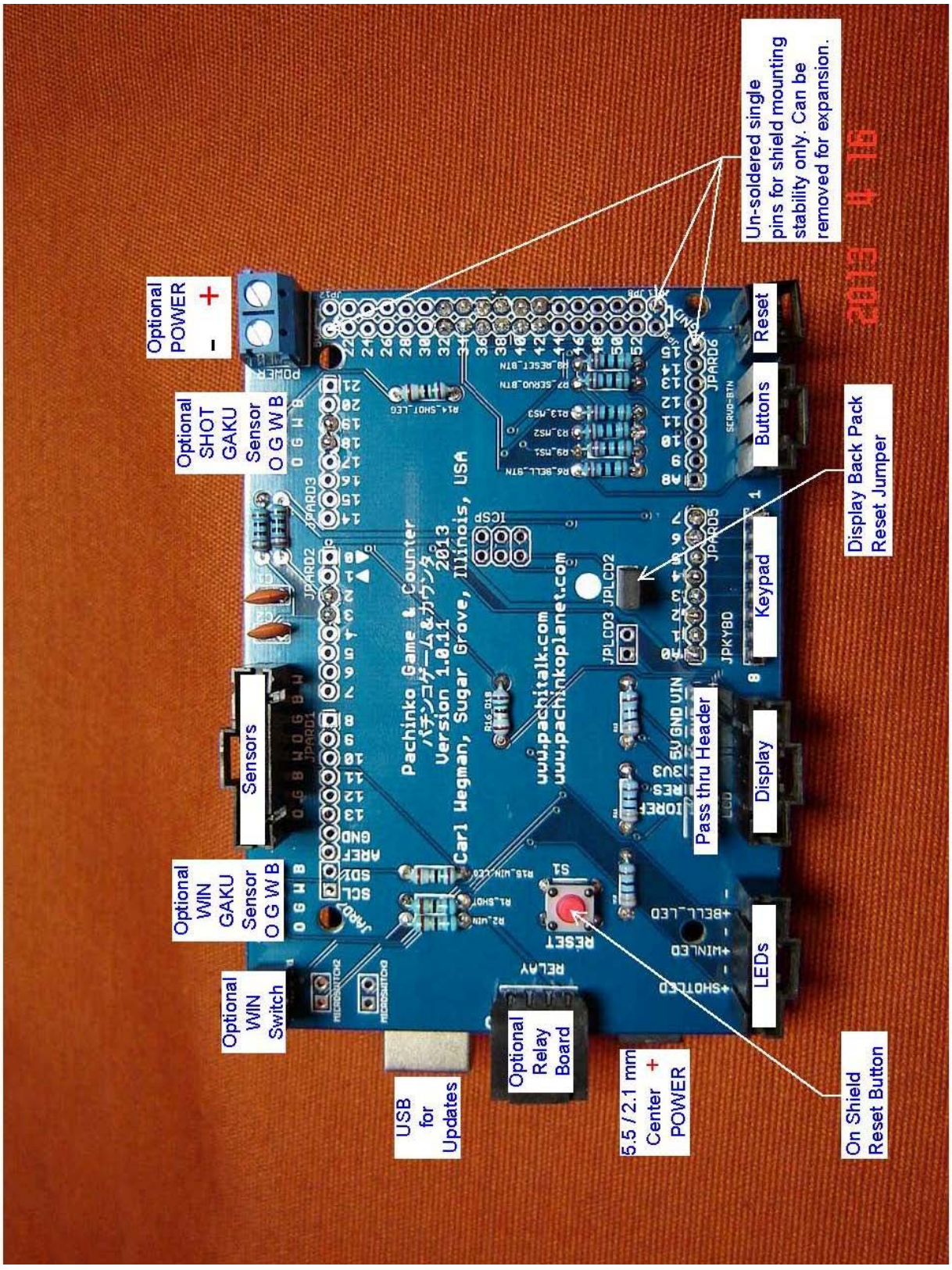
A. _____

ILLUSTRATIONS

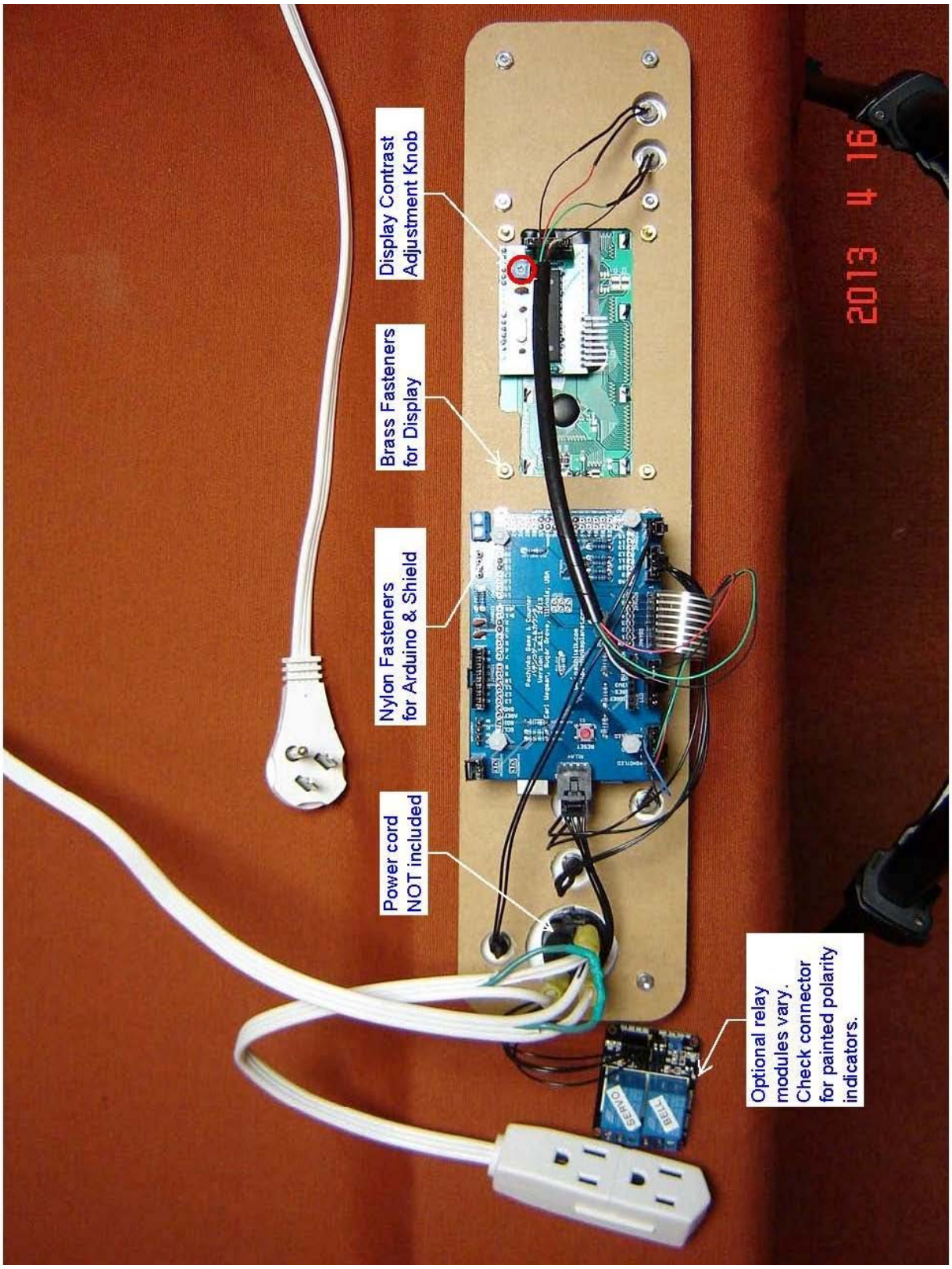
Connector Label	Description	Arduino PIN #	Polarized	Positions	Notes
A1	Win Micro Switch (optional)	35	Yes	2	
A2	Future Micro Switch 2	42	N/A	2	
A3	Future Micro Switch 3	43	N/A	2	
WL	Legacy (GAKU) Win Sensor	3	No	4	For GAKU Counter Win Sensor
S	Sensors	2,3	Yes	8	PINS 2 Shot, 3 Win
SL	Legacy (GAKU) Shot Sensor	2	No	4	For GAKU Counter Shot Sensor
R	Relays	39,40	Yes	4	Relay end of cable is not polarized PINS 39 Bell, 40 Servo
J1	Back Pack Jumper	N/A	N/A	2	Jumper not used
J2	RESET Jumper	N/A	N/A	2	Jumper USED
L	LEDs	36,37,38	Yes	6	PINS 36 Shot, 37 Win, 38 Bell
D	Display (4 x 20 LCD)	18,19	Yes	6	PINS 18 RX, 19 TX (SERIAL1)
		OR			
		32,33			PINS 32 RX, 33 TX
K	Keypad	A0 - A7	No	8	
B1	Pushbuttons	34, 41	Yes	4	PINS 34 Bell, 41 Servo
B2	RESET Pushbutton	N/A	Yes	2	

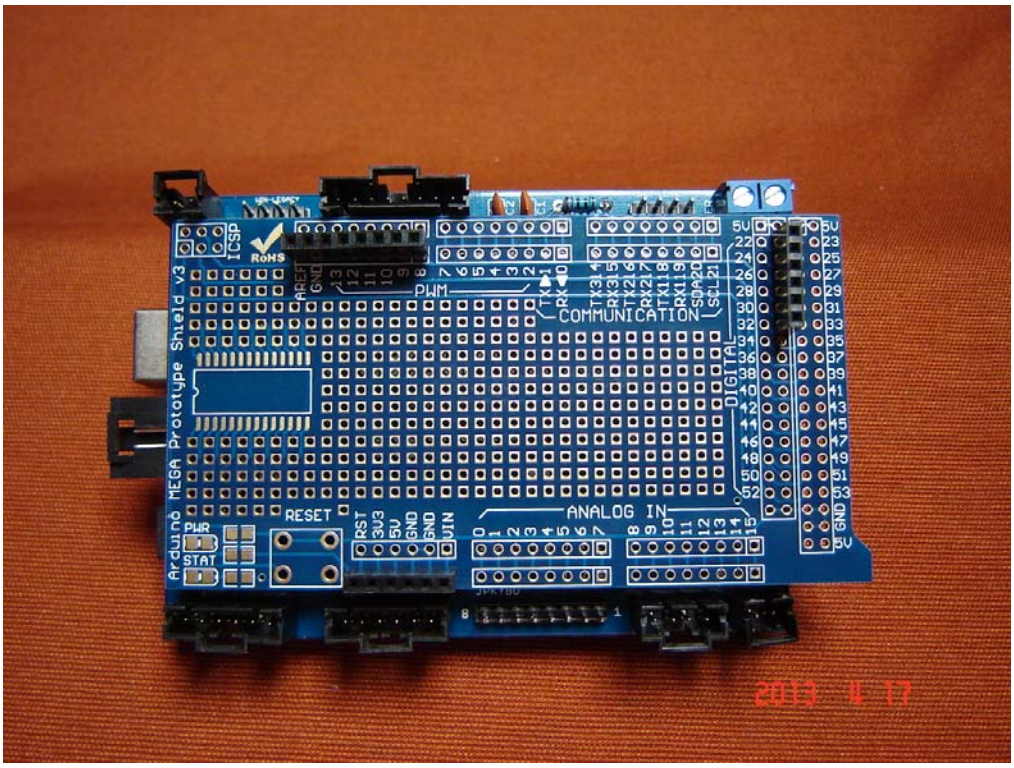


POWER 7-12V DC

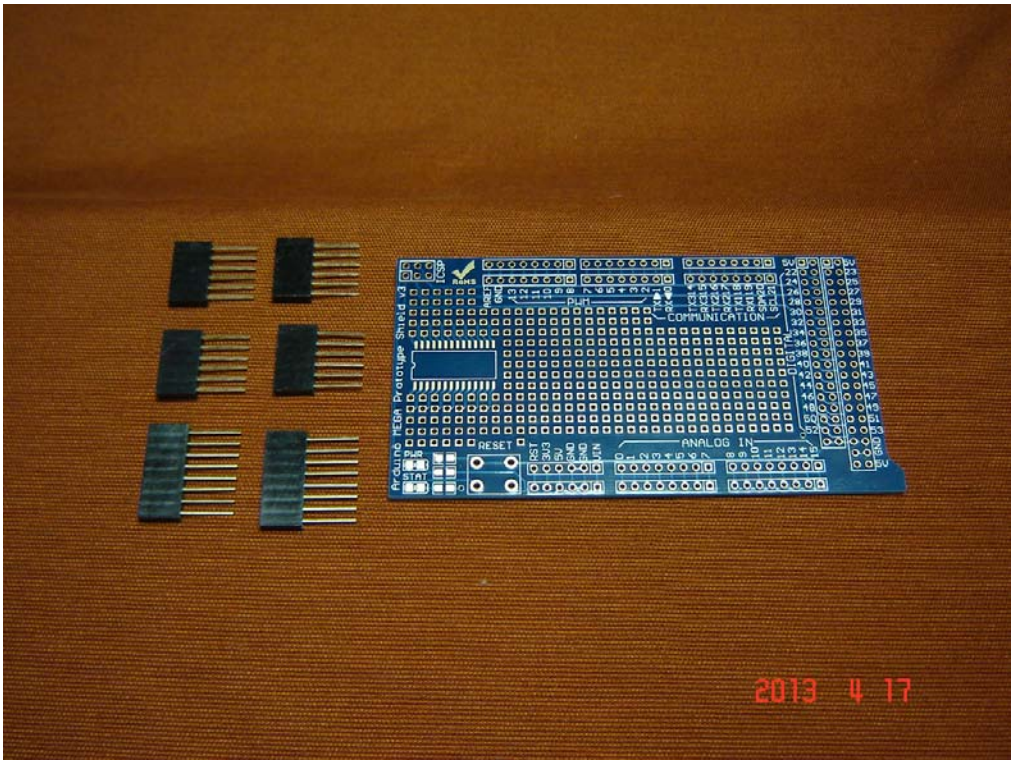


POWER 7-12V DC


Display Contrast Adjustment Knob
Brass Fasteners for Display
Nylon Fasteners for Arduino & Shield
Power cord NOT included
Optional relay modules vary. Check connector for painted polarity indicators.
2013 4 16

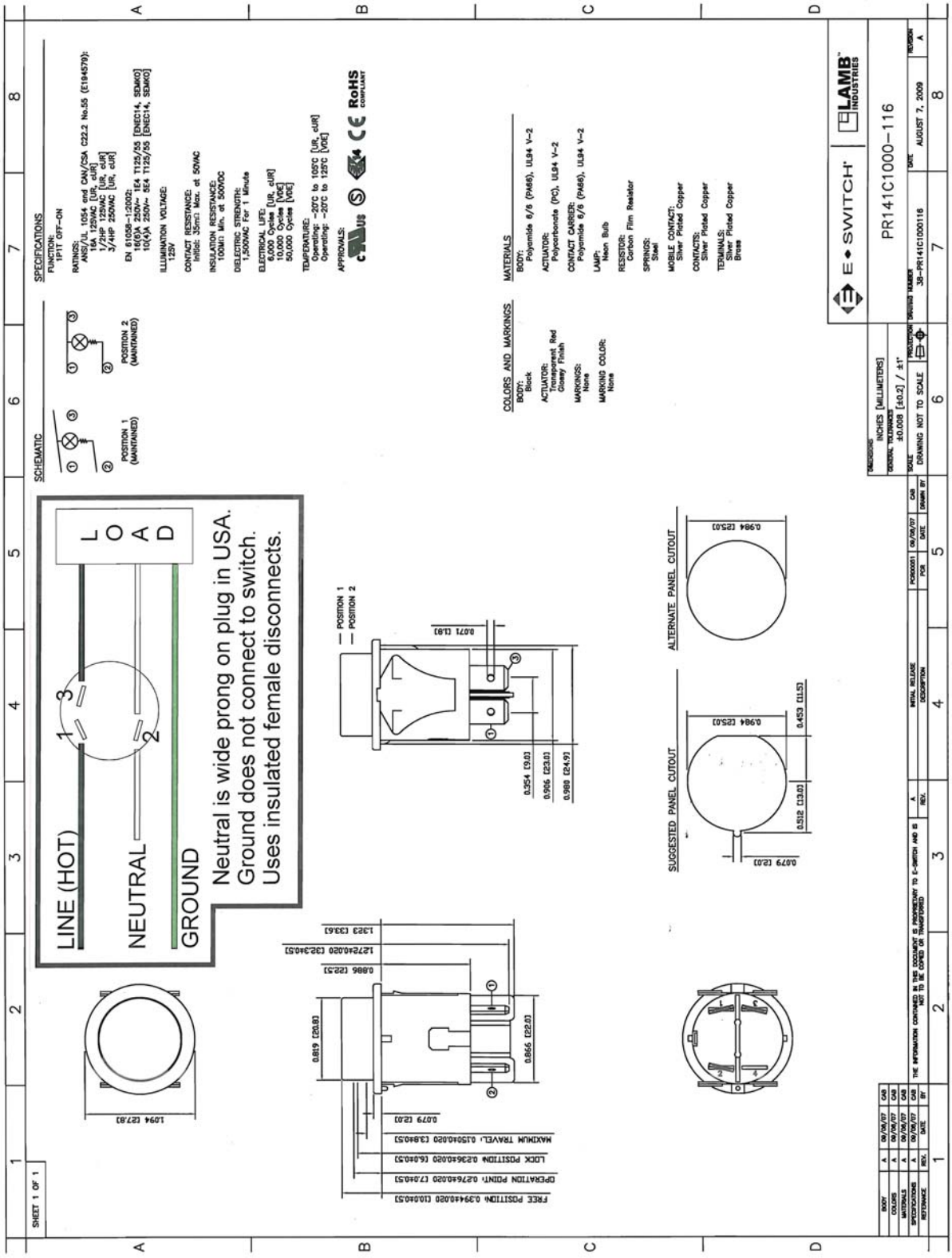


Assembled “PROTO” Shield Sample on top of Game



“PROTO” Shield Sample Parts

Version 11 May 2013


SPECIFICATIONS
SCHEMATIC
LINE (HOT)
NEUTRAL
GROUND
LOAD
Neutral is wide prong on plug in USA.
Ground does not connect to switch.
Uses insulated female disconnects.
POSITION 1 (MAINTAINED)
POSITION 2 (MAINTAINED)
COLORS AND MARKINGS
MATERIALS
SUGGESTED PANEL CUTOUT
ALTERNATE PANEL CUTOUT
E•SWITCH
LAMB INDUSTRIES
PR141C1000-116
AUGUST 7, 2009