Developers/ Integration Information V 2.07

SS2 Controller Versions:-
1.20 GLI Approved
1.21 Dragon Roulette
1.22 Bonus Number Output

CAMMEGH Roulette Wheels – Slingshot Series SS2
Slingshot 2 (SS2)
Slingshot TT (SS2 TT)

SS2 Series

Developers Information

www.cammegh.com

CAMMEGH
The World's Finest Roulette Wheel

About this guide

- This supplementary information has been developed for use with the models listed on the previous page only.
- This supplementary information is provided in addition to CAMMEGH SS2 Series Owners Handbook, and is intended for software developers/integrators or experienced personnel use only.
- All Cammegh Roulette Wheels have been precision engineered to the finest tolerances for consistent non-biased operation in the gaming environment.
- Before performing any maintenance or calibration work, if in doubt, consult the SS2 Series Owners handbook and appropriate wheel service booklet before attempting work.
- Should you require further assistance, please contact us at support@cammegh.com

Contents

Section 1 – Serial Port Configuration and Communications Protocol

1.1 – Serial Port Configuration

The SlingShot2 - Roulette Wheel Controller has two serial ports configured as follows:

Connector	Male 9-way Miniature 'D'
Pin Assignment	Identical to PC: Pin 2 = Receive; Pin 3 = Transmit; Pin 5 = Ground
Cable	To connect to a PC, use a Null Modem cable (3 wire female-to-female with pins 2, 3 and 5 connected, pins 2 and 3 being crossed over)
Baud Rate	9600
Data Bits	8
Parity	None
Stop Bits	1
Flow Control	None

1.2 – Communications Protocol

For the first few seconds after power on the SlingShot2 - Roulette Wheel Controller (controller) will not respond and the status LED will be 'Yellow'; during this time the controller is initialising.

If the LED does not change to green or flashing green, there is a hardware fault with the controller.

All packets to and from the controller begin with an asterisk (0x2A) and end with carriage return (0x0D, \r) *and/or* line feed (0x0A, \n), except the winning number reports, which do not start with an asterisk for backwards compatibility with display devices.

The controller is insensitive to which of \r and \n are used, but always terminates its packets by carriage return *and* line feed, \r\n.

Section 2 – Controller Information

2.1 - Normal Game Flow – Arcade Mode (default settings) - 1157

After power-up the controller is in idle mode and will automatically commence operation, with output of status information via the serial port(s). The controller will still respond to some user commands.

The following is an example of the communication involved in operating the wheel. The commands and responses are explained in detail in following sections:-

1) Controller is powered on and enters idle mode (Flashing Green LED).
2) Controller automatically starts rotating roulette wheel in arcade automatic mode.
3) The controller will start a game automatically.
4) The controller plays one game of roulette and goes back to (3), until powered down (5).
5) Controller receives ***P 0\r\n**, the wheel is stopped, and the controller enters idle mode (1).

2.2 - Normal Game Flow – Non Arcade Mode - 1159

After power-up the controller is in idle mode and will not move the wheel or output any status information via the serial port(s), however, the controller will respond to user commands.

The following is an example of the communication involved in operating the wheel. The commands and responses are explained in detail in following sections:-

1) Controller is powered on and enters idle mode (Flashing Green LED).
2) Controller receives ***K\r\n**, and replies with *K CAM\r\n.
3) Controller receives ***S\r\n**, and replies with *S <version number>\r\n (or use *V instead).
4) Controller receives ***F\r\n**, and wheel runs self-test and replies with *F <error code>\r\n (multiple times if several faults) then stops the wheel and re-enters idle mode.
5) Controller receives ***P 1\r\n**, and starts rotating roulette wheel in manual-start mode.
6) The controller will start a game when it receives either ***K\r\n**, **croupier key press** or ***u 1\r\n**.
7) Controller receives regular ***X\r\n** requests and replies with status reports.
8) The controller plays one game of roulette and goes back to (6), until powered down (9).
9) Controller receives ***P 0\r\n**, the wheel is stopped, and the controller enters idle mode (1).

For details of exception handling of the wheel see section 6.0.
The controller can also emulate an Automatic Cammegh In-Rim Reader (Mercury Wheel), see section 5.0.

Section 2 – Controller Information

2.3 – Controller Operating Mode

The command *(note < and > characters are never sent)*:

***o <*mode*>\r\n**

having the reply:

***o <*mode*>\r\n**

This command sets the operating mode of the controller. The operand *mode* is an ASCII decimal number sum of the desired option bit values chosen from the table below.

The Cammegh Slingshot **default** mode value is 1+4+128+1024 = **1157**
i.e. **Arcade Automatic Mode**.

For typical integration with game server/betting terminals, the mode value is 1+2+4+128+1024 = **1159**
i.e. **Non Arcade Mode**.

(Note that mode is stored in non-volatile memory and therefore need not be set on every power up once configured.)

Mode Option	Description	Mode Value
Reverse Wheel Each Game	Plays each game in the opposite direction to the last. If 0, each game is played clockwise.	1
User Game Start	Waits for external user key 1 press, *u 1\r\n or *K\r\n before starting the next game.	2
Modify Wheel Speed After No More Bets	Modifies the speed of the wheel by a random amount after No More Bets has been detected.	4
Disable Single Winning Number	Disables output of the single winning number (See section 2.4 Winning Number Output)	16
Automatic Status Output	Outputs the status of the wheel every 500 ms	128
Output *F Error Message During Play	Outputs *F errors during play. (See section 2.9 for details of error codes)	256
Move Game Delay	Move games per hour delay from state 5 to state 1.	1024
Move Game Delay (V1.21+ only)	Move games per hour delay from state 5 to state 2.	2048
Dragon Roulette Gameplay (V1.21+ only)	Enables Dragon Roulette Mode with gameplay + OP4 driver	4096
Sequential No. Sequence (V1.21+ only)	Enables Dragon Roulette Number (sequential)	8192
Bonus Number Output (V1.22+ only)	Enable Bonus Number Output	32768

Section 2 – Controller Information

2.4 – Winning Number Output

Optionally, when the controller detects a new winning number, the following is output:

<ASCII Number>\r\n

The number always occupies two characters; numbers 1-9 and the single zero are preceded by a space. This is the only packet sent by the controller that is not preceded by an asterisk (0x2A) and is required for compatibility with passive winning-number displays.

It may be switched off by setting bit 4 (+16) in the operating mode (see section 2.3).

Only serial port 1 is capable of outputting the winning number single line.

Section 2 – Controller Information

2.5 – Status Output – Advanced Protocol

The status is reported when a *X\r\n command is received (minimum of 100 ms between polls) or automatically every 500 ms if the operating mode has Automatic Status Output enabled. The format of the reply is (where < and > are not transmitted):

*X;<**State**>;<**Spin number**>;<**Winning number**>;<**Warning flags**>;<**Wheel speed**>;<**Wheel direction**>\r\n

State:
1 = Before game – Wheel starts moving after a close table state, or if used with operation mode 1024, duration will enforce the requested games per hour rate. (Green Light)
2 =Place your bets – Wheel is moving and a new game is being started. (Green Light)
3 = Ball In-Rim – Ball is moving in the rim track. Takes one revolution of the ball in the rim before detected. (Green Light)
4 = No More Bets – Ball to leave the rim track within approx. 3.5 revs, and in the opposite direction to the wheel. (Red Light)
5 = Winning number – Ball is detected in a pocket. (Yellow Light)
6 = Close table – Powered down (idle mode). (Flashing Green)

NOTE:
The normal flow of states is 6, 1, then an indefinite cycle from 2 through to 5.
If the wheel has a miss fire, it will stay in state 2 until the ball has been fired correctly.

Spin number:
The number of spins since power on. Range is 1 to 256 and wraps around. Padded to three characters.

Winning Number:
Last winning number in the same format (space-padded right-justified) as the winning number output (section 2.4). Before the first winning number is received (spin number 1) this field is a double space.

Warning flags:
Zero if no errors detected; non-zero indicates an error:
1 = The ball has been removed from the wheel.
2 = Ball is rotating in the same direction as the wheel during state 3 (swap rim air jets).
4 = Sensor(s) broken either on or off. Or the motor / encoder is faulty or disconnected.
8 = Failed ball launch; the ball did not reach the rim and the controller is retrying.

NOTE:
This field is a single hex digit, as code may add together. I.e. 8+2 = A

CONSULT SECTION 4.1 - EXCEPTION HANDLING for use of warning flags

Wheel Speed:
Current wheel rpm * 10. Padded to three characters.

Wheel Direction:
0 = clockwise
1 = counter clockwise

Section 2 – Controller Information

2.6 – Calibration Commands

For detailed procedure, please consult SS2 Series Owners Handbook Section 4 Diagnostics

The wheel requires two types of calibration: the positions of the sensors and the firing position.

NOTE:- Calibration position for single zero rotors = '0', for double zero rotors = '00'

To calibrate the sensor positions:

1) Power down the wheel (***P 0\r\n**, if not already not in idle mode).
2) Fix the ball in the middle of pocket '0' or '00' (Done by the pocket dimple).
3) Enter the command ***C\r\n**.
4) The controller replies with
   *C OK : Calibrate Sensor Positions (Ball In Middle '0')\r\n
5) After a few revolutions of the wheel, the controller will have finished calibration and reply with (Note: The rotations refer to the sensors wiring order, not wheel direction)
   *C BALL : OK Clockwise\r\n
   or
   *C BALL : OK Anti-Clockwise\r\n
   if the calibration was SUCCESSFUL, or
   *C BALL : No ball direction found\r\n
   if the calibration has FAILED.
6) If successful the controller will also reply with
   *C OK : Completed (<sensor 0 position>, <sensor 1 position>, <sensor 2 position>)\r\n
   and if unsuccessful will reply with
   *C FAIL : Sensor(s) <list of failed sensors> Broken\r\n

To calibrate the ball firing position:

1) Power down the wheel (***P 0\r\n**, if not already not in idle mode)
2) Rotate the wheel two revolutions clockwise by hand.
3) Continuing clockwise move pocket '0' (or '00'to the required firing position.
4) Enter the following commands to set, and then store to flash, the calibration:
   ***T c 20000\r\n**
   ***T f\r\n**

2.7 – Statistics Commands

The following commands are used to get statistics from the controller (only in state 5 or 6).

Command	Description	Reply
***w\r\n**	Get current Chi-squared value for the wheel.	*w <chi-squared> <as percent>\r\n
***W\r\n**	Get table of winning numbers. Only possible in states 5 and 6.	*W BUSY\r\n if in states 1, 2, 3 or 4. *W <total number of games>\r\n, then the list of pockets and number of times that each pocket has won.
***M\r\n**	Get table of misfired pockets. Only possible in states 5 and 6.	*M BUSY\r\n if in states 1, 2, 3 or 4. *M <total number of misses>\r\n, then the list of pockets and number of times that each pocket has misfired.
***s t\r\n**	Get number of number of used pages in winning number log.	*s t <pages>\r\n
***s r <page>\r\n**	Read page from winning number / misfire log. Only available in state 6 (idle/off). Winning Numbers (0-37) = 0x21 -> 0x46, Misfires (0-37) = 0x47 -> 0x6C, '~' = 0x7E	*s r <page> <256 bytes of data> <two character checksum>\r\n Data bytes are winning numbers (0-37), misfires (38-75) plus 0x21 to make printable ASCII; if Data is a tilde (~) then that byte is not used in the page. Checksum = Modulo 256 sum of all data bytes sent as a 2-character hexadecimal string.
***s e\r\n**	Erase all pages of log.	*s e ERASED WINNING NUMBER LOG\r\n

Section 2 – Controller Information

2.7 – Setup Commands

The following commands are used to set and get the controller's parameters. Note that *T commands do not store changes to flash memory immediately; *T f\r\n must be sent otherwise the changes will be lost on power off.

Italics = Inquiry Command **Bold** = Setting Command

Command	Description	Reply
***T d <distance>\r\n**	**Stopping distance before firing ball. Default = 15.**	***T OK\r\n**
**T D\r\n*	*Get stopping distance.*	**T d <distance>\r\n*
***T c <value>\r\n**	**Set the wheel position to be the firing position for pocket '0' (see section 3.4). If value is 20000 then the current physical wheel position is used as the firing position.**	**T OK\r\n**
**T C\r\n*	*Get the current wheel position for firing position for pocket '0'.*	**T c <position>\r\n*
***T p <time in ms>\r\n**	**Set the delay in mS between ball jet firing and rim jet firing. Default = 50ms**	***T OK\r\n**
**T P\r\n*	*Get the delay in mS between ball jet firing and rim jet firing.*	**T p <time>\r\n*
***T q <time in ms>\r\n**	**Set the delay in mS between rim jet firing and wheel starts to move. Default = 800ms**	***T OK\r\n**
**T Q\r\n*	*Get the delay in mS between rim jet firing and wheel starts to move.*	**T q <time>\r\n*
***T k <time>\r\n**	**Ball time for keeping the ball in the rim. Default = 1900. time = time for 1 revolution of the ball in the rim / 2048**	***T OK\r\n**
**T K\r\n*	*Get the ball time for keeping the ball in the rim.*	**T k <time>\r\n*
***T i <time>\r\n**	**Set the time (in 0.5 Sec steps) for the sensor broken off detection in state 4. Default = 10 (5Secs)**	***T OK\r\n**
**T I\r\n*	*Get the time (in 0.5 Sec steps) for the sensor broken off detection in state 4.*	**T i <time>\r\n*
***T j <sensor1> <sensor2> <sensor3> <sensor order>\r\n**	**Set the calibrated sensor positions. Must use *T f to store and switch the mains off and on again to use new settings.**	***T OK\r\n**
**T J\r\n*	*Get the calibrated sensor positions.*	**T j <sensor1> <sensor2> <sensor3> <sensor order>\r\n*

Section 2 – Controller Information

2.7 – Setup Commands

The following commands are used to set and get the controller's parameters. Note that *T commands do not store changes to flash memory immediately; *T f\r\n must be sent otherwise the changes will be lost on power off.

Italics = Inquiry Command **Bold** = Setting Command

Command	Description	Reply
***T r <min time> <max time>\r\n**	**Set minimum and maximum duration for in-rim air jets; a random value is selected for every game. Default = 28000, 32000.**	***T OK\r\n**
**T R\r\n*	*Get minimum and maximum duration for in-rim air jets.*	**T r <min time> <max time>\r\n*
***T s <min rpm> <max rpm>\r\n**	**Set minimum and maximum wheel speed; a random speed is selected for every game. Default = 14.0, 17.5**	**T OK\r\n**
**T S\r\n*	*Get minimum and maximum wheel speed*	**T s <min rpm> <max rpm>\r\n*
***T t <distance>\r\n**	**Set wheel deceleration distance to firing position. Default = 4000. (Maximum = 9999 = 1Rev)**	***T OK\t\n**
**T T\r\n*	*Get wheel deceleration distance to firing position*	**T t <distance>\r\n*
***T n <time>\r\n**	**Set ball rotation time for the detection of no more bets. Default = 2920. time = time for 1 revolution of the ball in the rim / 2048**	***T OK\r\n**
**T N\r\n*	*Get ball rotation time for the detection of no more bets*	**T n <time>\r\n*
***T m <rpm>\r\n**	**Set the maximum amount by which the speed of the wheel Is modified after no more bets (if enabled) Default = 1.5 i.e. -1.5 to +1.5 RPM**	***T m <rpm>\r\n**
**T M\r\n*	*Get rotor speed modification RPM after no more bets*	**T m <rpm>\r\n*
***T f\r\n**	**Store the configuration values to flash**	***T OK\r\n**
***T h <games per hour>\r\n**	**Set the maximum number of games per hour that the wheel will play. Default = 60. The wheel will stay in state 5 for the required time to enforce the requested games / hour.**	***T OK\r\n**
**T H\r\n*	*Get the maximum number of games per hour*	**T h <games per hour>\r\n*
***T z <num of pockets>\r\n**	**Inform the controller of the number of pockets on the wheel, either 37 or 38 (double zero wheel). Default = 38 (double zero setting)**	***Z OK\r\n**
**T Z\r\n*	*Get the number of pockets*	**T z <pockets>\r\n*

Section 2 – Controller Information

2.8 – Power, Status and Time Commands

Command	Description	Reply
*P <0=Off, 1=On>\r\n	Set power state.	*P OK\r\n
*R\r\n	Reset state back to state 1 (forcing game restart).	*R OK\r\n
*S\r\n	Get firmware version number.	*S <version number>\r\n
*V\r\n	Get full firmware version.	*V <version string>\r\n
*X\r\n	Get status report.	*X;…\r\n
*K\r\n	Get manufacturer code/start game.	*K CAM\r\n (manufacturer, in idle mode) *K OK\r\n (start game)
*u 1\r\n	Start a new game in manual mode.	*u OK\r\n
*I 4\r\n	Output sensor detection events in the format:- bs n \r\n bn nn \r\n Where n = sensor number, nn = ball in pocket number	*I 4 \r\n *bs n \r\n *bn n\r\n
*I 0\r\n	Cancel sensor output events	*I 0 \r\n
*A\r\n	Report the current wheel position as the number of pockets since the last datum point on the encoder.	*A <ASCII number>\r\n
*d g\r\n	Get Time and Date from Real Time Clock	*d g HH:MM:SS – DD/MM/YY\r\n

*d s <select> <val>\r\n	Set Date / time where:-
	Select = 0 Set Seconds
	Select = 1 Set Minutes
	Select = 2 Set Hours
	Select = 3 Set Day
	Select = 4 Set Month
	Select = 5 Set Year
		*d OK\r\n

Section 2 – Controller Information

2.9 – Self-test Command

The controller can perform a self-test. From idle mode, issue the following command:

***F\r\n**

If the controller is not in the idle state (6) it will reply with

*F BUSY\r\n

The controller will take a maximum of 5 minutes to perform the test. If no errors are detected the controller will reply with *F 0\r\n. Otherwise, the controller will reply with one or more *F packets, each with an error code. The following table lists the error codes.

Error Code	Description
0	No errors detected
1	Hardware fault
2	No ball detected
3	No ball position detected
4	Detected ball direction is not correct
5	Motor is not driving the wheel correctly
6	Encoder has failed (or wheel hasn’t been calibrated yet)
7	Could not play a game clockwise: no air jets or no calibration yet
8	Could not play a game anti-clockwise: no air jets or no calibration yet
11	Sensor 0 broken On (Stuck On)
12	Sensor 1 broken On (Stuck On)
13	Sensor 2 broken On (Stuck On)
21	Sensor 0 broken Off (Cannot see ball)
22	Sensor 1 broken Off (Cannot see ball)
23	Sensor 2 broken Off (Cannot see ball)
100	Test has timed out

2.10 – Setup / Test Commands

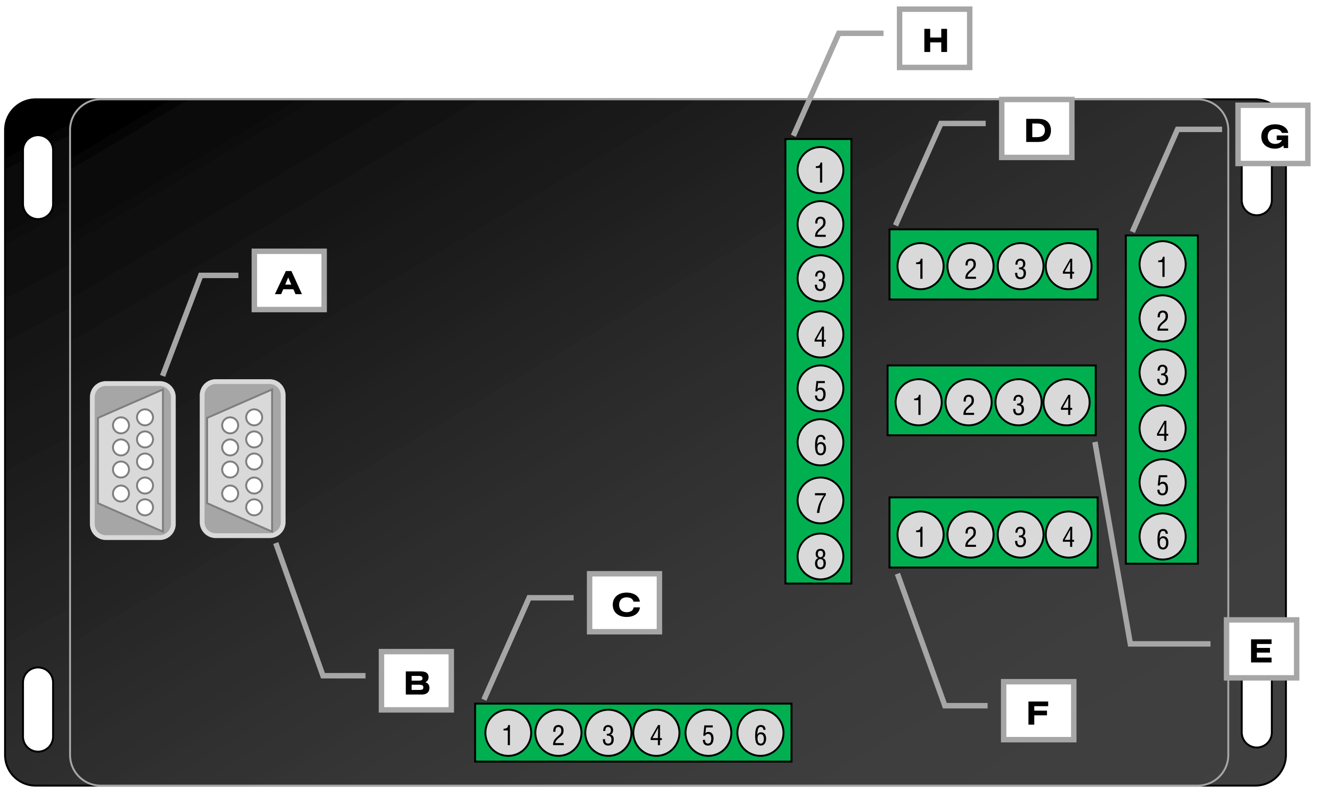
The following commands can be used to setup/test the controller.

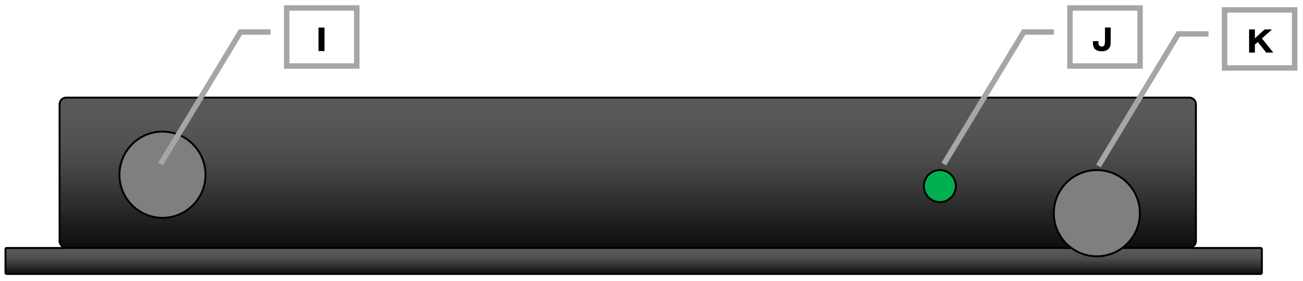
Command	Description	Reply
*i a <time>\r\n	Setup all pockets to fire the ball out with a time of value.	*i all pockets = <time>\r\n
*i g <pocket_no>\r\n	Returns the current ball fire time for requested pocket.	*i <pocket_no>=<time>\r\n
*i s <pocket_no> <time>\r\n	Setup requested pocket to the requested time.	*i <pocket_no>=<time>\r\n
*i f\r\n	Store all pocket times to flash.	*i stored\r\n
*j <direction> <inrim_time> <pocket_time> <repeat_delay>\r\n	Direction = 0 for disable feature, 1 anti-clockwise jet, 2 clockwise jet. inrim_time = on time for inrim jet, pocket_time = pocket puffer jet time, repeat_delay = time between fires times 2.	*j OK\r\n
*O\r\n	Get extra setup parameters. This is used by the Windows stats program.	*O <wheel state> <enable_protocol> <Games Per Hour> <Operating Mode>\r\n
*q <enable>\r\n	Used to switch off *X protocol so you can read other debug output. enable 0=disable ANY *X protocol output, 1=enable protocol (normal operation)	*q OK\r\n

Section 3 – Wiring Diagram

3.1 – SS Game Controller Wiring Connections

The following table details the connectors and pin-outs for the controller.





NOTES:

1) The order of the sensors does not matter, however if changed the sensor positions will need re-calibrating.
2) Serial Port 2 (B) is only used for displays, as it doesn't support the statistics commands (except *w), *l debug commands and the *C sensor position calibrating command.
3) WARNING: Make sure that the motor and encoder are plugged into the correct position, as they are both 6-way connectors.
4) User Key inputs are activated on short to 0V. I.e. Wire switch button between User Key input and 0V

Section 3 – Wiring Diagram

3.1 – SS Game Controller Wiring Connections

Connector	Description	Pin Number	Pin Detail
A	Serial Port 1	**-**	-
B	Serial Port 2	**-**	-
C	Stepper Motor	**1**	No Connect
		2	Red
		3	Yellow
		4	Blue
		5	Orange
		6	No Connect
D	Sensor 0	**1**	Brown
		2	No Connect
		3	Black
		4	Blue
E	Sensor 1	**1**	Brown
		2	No Connect
		3	Black
		4	Blue
F	Sensor 2	**1**	Brown
		2	No Connect
		3	Black
		4	Blue
G	Encoder	**1**	Black *and* Shield
		2	Orange
		3	Red
		4	Brown
		5	White
		6	No Connect
H	Air Solenoids	**1**	Jet3 – Under Wheel Jet (Puffer Jet)
		2	Jet3 – Under Wheel Jet (Puffer Jet)
		3	Jet2 – Clockwise In-Rim Jet
		4	Jet2 – Clockwise In-Rim Jet
		5	Jet1 – Anti-Clockwise In-Rim Jet
		6	Jet1 – Anti-Clockwise In-Rim Jet
		7	No Connect
		8	No Connect
I	Power Input	**1**	0V
		2	Ground
		3	+24V
		4	0V
		5	+24V
J	Light (LED)	**-**	-
K	User Input And External Light (LED)	**1**	Light (LED1)
		2	Light (LED2)
		3	0V
		4	User Key 1 (Manual Game Start)
		5	User Key 2 (Spare)
		6	0V

Section 4 – Exception Handling

4.1 – Exception Handling

The controller can operate in many different modes however the status protocol is still the same, although you may have to request it instead of receiving it at regular intervals. This section assumes the developer has already selected their operating mode from section 2.1 and understands the status protocols detailed in section 2.3.

Exception handling is performed using the received status protocol. The controller detects three types of exception.

Exception	Description	User Detection Method / Action
Failed ball launch	The ball did not reach the rim and the controller is retrying.	**Warning flag code 8** is set and the controller will retry. No action is required the warning flag will clear when the ball is successfully launched, the wheel state also change to 3 (ball in rim). It is feasible that warning flag 8 is IGNORED for the first 3 seconds of State 3, unless the warning flag persists beyond this period of time, in which case either the wheel controller will automatically try to re-launch the ball, or the developer should VOID GAME or STOP GAME to investigate
Miss-Spin	The ball was launched correctly, however has not completed a normal spin, i.e. ball has hit an object.	The wheel state is 3, and **Warning flag code 2** is set. This is a VOID GAME, the developer must LOCK all bets and send the wheel a restart game command (*R\r\n) thereby acknowledging the VOID GAME. The wheel will now re-spin the ball from state 2 and play will continue.
No ball detected	The ball was launched correctly, however has not been detected by the in-rim sensors for a period of one complete revolution of the rotor	The wheel state is 4, and **Warning flag code 1** is set. This warning flag should be IGNORED for a minimum period of 5 seconds (10 x messages @ 500ms output protocol). If the warning flag persists beyond this time period, the developer should VOID GAME or STOP GAME to investigate. The wheel will automatically stop if the warning flag 1 persists for 75 seconds or more.
Component fault (Sensor, Motor or encoder fault)	The wheel has detected that one or more of its component(s) has failed	If the operating mode has flag 256 enabled the wheel will output one or more *F <error code>\r\n error codes at the point of fault detection. If the fault is permanent the wheel will then power down. If the operating mode doesn't use flag 256 then the wheel will power down without warning (i.e. wheel will stop rotating and enter state 6). The developer should then perform a built in self-test to re-detect the fault. Error code 4 will also indicate the fault.

Section 5 – Arcade Mode Use

5.1 – Cammegh Slingshot 2 Arcade Mode

The Cammegh Slingshot 2 can be operated in an 'Arcade' mode, operating full automatically based on the configuration of the game parameters to achieve desired gameplay characteristics.

5.2 – Cammegh Slingshot 2 Arcade Mode Parameters

To operate in Arcade (fully Automatic) mode, the following parameters will be applicable:-

Operation mode (as per section 2.3 – page 5)

- This setting will determine the operation mode of the wheel, for arcade mode, typically the value is 1157.
- If operation mode value 1024 is used, the wheel will stay in state 5 for 2 rotor revolutions and return to state 1 where the length of state 1 will enforce the requested games per hour rate.
  The game protocol will be as per example 5.3 – page 19.
- If the operation mode 1024 value is NOT used, the wheel will stay in state 5 for the required time to enforce the requested games per hour rate, and return to state 2.
  The game protocol will be as per example 5.4 – page 20.

Games per hour setting - GPH (as per section 2.7 – page 11)

- This setting will determine the average games per hour operation rate.
- Check current setting *T H <enter>
- Change command *T h x <enter> where x represents the requested games per hour rate.
- The default setting = 60 (GPH)
- The setting must be saved to flash memory by confirming with the command *T f <enter>

No More Bets – NMB Threshold detection setting (as per section 2.7 – page 11)

- This setting will determine when the state 3 (ball in rim) to state 4 (no more bets) transition occurs.
- Check current setting *T N <enter>
- Change command *T n xxxx <enter> where xxxx represents time setting as below.
- The value / 2048 = time for 1 revolution of the ball in the rim.
- The default setting = 2920 (2920/2048 = 1.4 seconds or 42.8 rpm)
  i.e. when the ball has reached a speed of 42.8 rpm or a rotation time of 1.4 seconds, the NBM threshold will be reached.
- Increasing the value will decrease the length of state 4
  i.e. NMB will be called later in the game / closer to ball drop.
- Decreasing the value will increase the length of state 4
  i.e. NMB will be called earlier in the game / closer to ball launch.
- The setting must be saved to flash memory by confirming with the command *T f <enter>

Section 5 – Arcade Mode Use

5.3 – Protocol Example – Using operation mode value 1024

If the operation mode includes the use of the 1024 setting (Move game delay from state 5 to state 1) the game protocol will follow the sequence below.

(For detailed protocol explanation see section 2.5 – page 7)

***X;1;011;23;0;154;0**	***State 1 – Wheel Started / Ready for Start***
*X;1;011;23;0;154;0	*Wheel idling with delay to achieve requested GPH rate.*
*X;1;011;23;0;154;0	
*X;1;011;23;0;154;0	
***X;2;011;23;0;154;0**	***State 2 – Wheel ready for Start / Launching Ball***
*X;2;011;23;0;150;0	*Controller signalled to start game.*
*X;2;011;23;0;136;0	*Wheel finds ball position and decelerates to firing position.*
*X;2;011;23;0;086;0	*Time period will vary depending on degrees of rotation to firing position.*
*X;2;011;23;0;042;0	
***X;3;011;23;0;170;1**	***State 3 – Ball launched and detected in ball track***
*X;3;011;23;0;170;1	*Ball spinning around ball track*
*X;3;011;23;0;170;1	*(recommend ignore error 8 for first 3 seconds of state 3 in case of*
*X;3;011;23;0;170;1	*false triggers at initial ball launch revolution)*
*X;3;011;23;0;170;1	
***X;4;011;23;0;168;1**	***State 4 – Ball decelerated to NMB threshold***
*X;4;011;23;0;165;1	*RRS modification of rotor speed (if operation mode 4 is enabled)*
*X;4;011;23;0;165;1	*Waiting for ball to drop*
*X;4;011;23;0;165;1	
*X;4;011;23;0;165;1	
25	***W/N Output (if enabled in operation mode + connected to serial 1)***
***X;5;011;25;0;165;1**	***State 5 – Winning number detected***
*X;5;011;25;0;165;1	*3 sensors agree ball in pocket position*
*X;5;011;25;0;165;1	*State 5 length = 2 complete revolutions of the rotor.*
*X;5;011;25;0;165;1	
*X;5;011;25;0;165;1	
***X;1;011;25;0;165;1**	***State 1 – Wheel Started / Ready for Start***
*X;1;011;25;0;165;1	*Wheel idling with delay to achieve requested GPH rate*
*X;1;011;25;0;165;1	

Sequence shortened for illustrative purposes – Game State Loops 1,2,3,4,5

Section 5 – Arcade Mode Use

5.4 – Protocol Example – NOT Using operation mode value 1024

If the operation mode does NOT includes the use of the 1024 setting (Move game delay from state 5 to state 1) the game protocol will follow the sequence below.

(For detailed protocol explanation see section 2.5 – page 7)

***X;1;011;23;0;154;0**	***State 1 – Wheel Started / Ready for Start***
*X;1;011;23;0;154;0	*Only seen after wheel power up / first game*
*X;1;011;23;0;154;0	*Duration = 4 revolutions.*
*X;1;011;23;0;154;0	
***X;2;011;23;0;154;0**	***State 2 – Wheel ready for Start / Launching Ball***
*X;2;011;23;0;150;0	*Controller signalled to start game.*
*X;2;011;23;0;136;0	*Wheel finds ball position and decelerates to firing position.*
*X;2;011;23;0;086;0	*Time period will vary depending on degrees of rotation to firing position.*
*X;2;011;23;0;042;0	
***X;3;011;23;0;170;1**	***State 3 – Ball launched and detected in ball track***
*X;3;011;23;0;170;1	*Ball spinning around ball track*
*X;3;011;23;0;170;1	*(recommend ignore error 8 for first 3 seconds of state 3 in case of*
*X;3;011;23;0;170;1	*false triggers at initial ball launch revolution)*
*X;3;011;23;0;170;1	
***X;4;011;23;0;168;1**	***State 4 – Ball decelerated to NMB threshold***
*X;4;011;23;0;165;1	*RRS modification of rotor speed (if operation mode 4 is enabled)*
*X;4;011;23;0;165;1	*Waiting for ball to drop*
*X;4;011;23;0;165;1	
*X;4;011;23;0;165;1	
25	***W/N Output (if enabled in operation mode + connected to serial 1)***
***X;5;011;25;0;165;1**	***State 5 – Winning number detected***
*X;5;011;25;0;165;1	*3 sensors agree ball in pocket position*
*X;5;011;25;0;165;1	*Wheel idling with delay to achieve requested GPH rate.*
*X;5;011;25;0;165;1	
*X;5;011;25;0;165;1	
***X;2;011;25;0;165;1**	***State 2 – Wheel ready for Start / Launching Ball***
*X;2;011;25;0;165;1	*Controller signalled to start game.*
*X;2;011;25;0;165;1	

Sequence shortened for illustrative purposes – Game State Loops 2,3,4,5

Section 6 – Bonus Number Output

6.1 – Enable Bonus Number Output

- Requires controller version 1.22 minimum.
- Enable bonus number output using SS2 operating mode value – see page 5
- Bonus number output is an expiration licensed feature, based on time period of use.
- The license period is set in days (24 hour power on periods). This period will reduce by 1 unit for every power on of the system.

6.2 – Bonus Number Output Protocol

The following lines are output in addition to the standard protocol – see page 7

6.2.1 – Bonus Numbers Bonus Number 1
At the start of State 3 (after the first protocol line in this state), a line in the format:-
"*Y;1;nnn\r\n"
Where nnn represents a bonus number (1) in the range 000 to 255

6.2.2 – Bonus Numbers Bonus Number 2
At the start of State 3 (after the bonus number 1 line), a line in the format;-
"*Y;2;nnn\r\n"
Where nnn represents a bonus number (2) in the range 000 to 255

6.2.3 – Bonus Numbers Bonus Number 3
At the start of State 4 (after the first protocol line in this state), a line in the format;-
"*Y;3;nnn\r\n"
Where nnn represents a bonus number (3) in the range 000 to 255

6.2.4 – Bonus Numbers Bonus Number 4
At the start of State 5 (after the first protocol line in this state and single winning number line), a line in the format;-
"*Y;4;nnn\r\n"
Where nnn represents a bonus number (4) in the range 000 to 255

6.2.5 – Bonus Numbers Bonus Number 5
At the start of State 5 (after the bonus number 4 line), a line in the format;-
"*Y;5;nnn\r\n"
Where nnn represents a bonus number (5) in the range 000 to 255

6.2.6 – Bonus Numbers Bonus Number 6
At the start of State 4 (after the bonus number 3 line), a line in the format;-
"*Y;6;nnn\r\n"
Where nnn represents a bonus number (6) in the range 000 to 255

NOTE:- In the event that there is 30 days or less bonus number output licenses remaining, the number of days remaining will be appended to the end of the *Y bonus number lines. Eg *Y;1;123;30, *Y;1;123;29...

If the bonus numbers license expires, all bonus numbers will have value 999. E.g. *Y;1;999;00, *Y;2;999;00...

Section 6 – Bonus Number Output

6.3 – Bonus Number Output Commands

Command	Description	Reply
*y\r\n	Returns the number of license days/power cycle remaining.	*y nnn \r\n
*y R\r\n	Get license request code for license renewal.	*y R xxxx-xxxx-xxxx-xxxx \r\n
*y A xxxx-xxxx-xxxx-xxxx\r\n	Input new license code giving nnn number of license days	*y A OK(nnn) \r\n
*o nnnnn \r\n	Operating mode should be set at normal use value + value 32768	*o nnnnn r\n

Section 7 – Firmware Version Change Notes

7.1 – Cammegh Slingshot 2 Controller Firmware Change Notes

The following notes apply following changes between the listed FW versions;-

Original version - v1.20 – GLI Approved

New release version - v1.22

7.2 – Developers Change Notes

Change Ref	Description
1.21.1	Change Version Number from 1.20 to 1.22
1.21.2	Added *T O and o commands for Dragon Wheel idle time (Dragon wheel mode only) Wheel speed during the result detected idle time (in state 4), value in RPM.
1.21.3	Added *T W and w commands for Dragon Wheel minimum launch delay. (Dragon wheel mode only) Pause wheel at fire position time delay on start of new game, value in 0.5 secs.
1.21.4	Added external flash writing code for the two above variables.
1.21.5	Changed the default for maximum wheel idle rpm speed to 17.5 RPM.
1.21.6	Changed the default for delay between puffer fire and rim fire to 50ms.
1.21.7	Changed the default for error timeout for state 4 to 10 seconds.
1.21.8	Fixed a bug in the Motor Power up command, due to new datasheet update.
1.21.9	Added support for Dragon Wheel to control the 4th output. 4th output = relay control for smart glass switching.
1.21.10	Added support for the dragon wheel number sequence (both single and double zero). *o operating mode defined.
1.21.11	Added support for moving the game delay from state 5 to state 2 at launch time (for dragon wheel mode). *o operating mode defined.
1.21.12	Added support for Dragon Glass control and state machine changes to handle the wait at launch position.
1.21.13	Changed to LED status colours;- Green; Ready for Button Press No Illumination;- Red; No More Bets Red Flashing; - Ready to release winning number Amber; Release winning number.
1.22.1	Added bonus number output code, operation mode activation and BN OP licensing system.

No changes to the FPGA code, nor the RPG code have been made

Appendix Note

V2.01 – Publication of new document version.

V2.02 – Publication layout changes minor.

V2.03 – Correction of compilation error section 2.3 mode value 64 corrected to 128, section 2.6 command corrected from *TC 20000 to *Tc 20000.

V2.04 – Addition of Section 5 – Arcade Mode Use information for system integrators. correct compilation reference errors

V2.05 – Addition of Section 6 – Developers change notes added for firmware update.

V2.06 – Addition of Section 6 – Bonus Number Mode . Move firmware change notes to section 7.

V2.07 – Extra information on Exception Handling Section 4.1 updated for use of warning flags. Unified documents to cover all controller versions, 1.20 GLI Approved, 1.21 Dragon Roulette Compatibility, 1.22 Bonus Number OP.