Content

1 Controller Types....2
1.1 Auto-ranging....2
1.2 Auto-Hold....2
2 EHC_CR: External Hardware Controller Configuration Register....4
2.1 Table of Hardware Source IDs....5
2.2 Table of ADC Channels....5
3 EHC_SR: External Hardware Controller Status Register....6
4 Controller Assignment....7
5 EHC Control Commands (LPC Messages)....8
5.1 Enable/Disable....8
5.2 Ranging....8
5.3 Forcing Output Update....8
5.4 Reset or Delete Controller....8
5.5 Get EHC Data....9
5.6 Save EHC Data to EEPROM....9
6 Auto-Save/Auto-Restore to/from EEPROM....10
7 Compatibility....11

1 Controller Types

An external hardware controller can either be a *3-wire* type or a *2-wire* type.

3-wire type is typically a passive *potentiometer*, and potentiometric readout is used, that is, the *ratio* of the wiper voltage and the top end voltage is calculated. The wiper contract is read out unloaded, the top end contact is connected to a 5V supply via a 10kOhms pullup.

2-wire types can be passive adjustable resistors (rheostats), passive mechanical/optical/analog switches (to Ground) or active control voltage sources. Control voltages are read out unloaded whereas rheostats and switches use a 10kOhms pullup to a 5V supply to generate an input voltage. Rheostatic readout is done via calculating the approximate resistance from the voltage divider equation which establishes a pretty much linear output characteristic for a linear resistance change. Control Voltages are read out directly as ADC values with no conversion.

Controller final output values can be either *continuous*, spanning a continuous range of values somewhere inside a 0%...100% interval, or they can be *bi-stable*, allowing only two output values, 0% and 100%. Bi-stable controllers first determine a temporary output value like a contionuous controller but further apply thresholds with hysteresis for determining the final output value.

Controller final output values can be *inverted*.

1.1 Auto-ranging

Auto-ranging can be enabled for any type of controller. When enabled, current minimum and maximum values are used to normalize the output to 0%...100% even then the physical input only covers a small- to medium-sized span of values. Further, actual min/max limits are backed off a little bit for normalization to ensure 0% and 100% always can be reached even when the controller raw input value is noisy and/or doesn’t always reach the same end-stop values. To dial in the Auto-Ranging, move the controller through its whole physical travel and back again, emphasizing on the end stops, remaing there for something like a second.

The back-off for the lower range end is 4%, and the back off for the upper range end is 5%. Those are empirical values to just have enough back-off for most commercial expression and damper pedals to work with auto-ranging. Too much back-off would restrict active travel of a controller without need. In rare cases where the back-off is not sufficient, manual ranging still can be used with range ends set to useful values.

For best results, expression pedals which have “mimimum” or “range” adjusts, those should be set to achieve the largest possible span of output (set “minimum” to “0”, or “range” to “max”). Controllers in 2-wire mode are especially sensitive to noise/instability when their actual value range is small.

1.2 Auto-Hold

Auto-hold is a mechanism to „freeze“ the current output of a continuous controller when its output would result in only very small amounts, until a large enough change – drift or intentional movement – has been detected again. It can be set in 5 steps of strength from 0(disabled) to 4(extreme). This is useful to de-noise and stabilize controllers that a sharing a macro control on which they operate together with other controllers (internal and external). By this, the normal noise from a controller does not overwrite the macro control output and therefore a value selected by, for example, a ribbon control remains valid and only gets overridden by the external controller when that controller detected enough movement. The algorithm is self-adapting dynamically, that is, as long as the controller moves fast, the output is never frozen, freezing only happens when the controller is changing only little for some time period. To come out of freezing, a somewhat larger movement is required, establishing a sort of hystersis.

Auto-Hold normally should not be disabled to avoid frequent “noise updates”. On the other hand, setting it to unnecessary strong levels reduces the apparent resolution that can be achieved when moving the controller slowly as the output values change in larger discrete steps. The correct setting depends strongly on the hardware, some expression pedals have been found to be very unstable and drifty compared to others. It also depends on the physical range spanned by the controller, some pedals have “minimum” or “range” adjustments and when those are reducing the effective span of the controller, a higher Auto-Hold setting might be needed.

It is recommended that the Playgound temporarily increases the user-selected Auto-Hold strenght by one level when it detects that a controller is used together with others (internal or external) on a shared Macro Control. This increase should be applied to all associated external controllers and it should be undone when it is detected that the controller will be the only single source for a Macro Control (or not linked to any Macro Control at all). The same temporary increase should be used when several controllers are set up to use the same HWSID.

Auto-hold is ignored for bi-stable controllers.

2 EHC_CR: External Hardware Controller Configuration Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Description	HWSID[3:0]				SIL	CTRLID[2:0]			POT	PUP	INV	ARAE	CONT	AHS[2:0]		

EHC_CR is accessed by sending LPC message `0x0F00, 0x0002, 0x0100, EHC_CR` (LPC message "BB_MSG_TYPE_EHC_CONFIG" with command ID "EHC_COMMAND_SET_CONTROL_REGISTER").

HWSID[3:0] : The *Hardware Source ID* (0..11, 15) to be used for message transmission to Audio Engine and Playground. See table „Hardware Source IDs" for meaningful IDs. HWSID can be updated at any time and does not trigger a full hardware init of the controller. **ID 15** has a special meaning, it is used to disable a controller entirely, no processing of that controller is done and the controller is removed from processing and all its internal data is cleared.

Note: *It is perfectly valid to assign HWSIDs that are already associated to non-EHC control devices like the built-in pitch bender, aftertouch and ribbons*. By this, an EHC can take over the role of an internal control source, working in parallel. Also, the same HWSID can be assigned to multiple controllers (see "Controller Assignment").

SIL : *Silent Mode*. When set, the Hardware Source ID is not transmitted to Audio Engine. This might be useful to implement triggered control flow actions („next preset", etc) assigned to controllers. SIL can be updated at any time and does not trigger a full hardware init of the controller.

CTRLID[2:0] : The *Controller ID* (0…7) for which the configuration will be applied. This is also the ID of main ADC channel involved. See table „ADC channel IDs".

POT : Indicates attached hardware is treated as a 3-wire standard *Potentiometer*. Selecting POT makes the controller automatically use the adjacent ADC channel on the same TRS jack with pullup to apply the control voltage (and measure the voltage, for the potentiometric ration calculation). Changing POT will cause a full hardware init of the controller, also resetting (auto-)ranging.

PUP : *Pullup* is applied to the input line. Should be set for reading out passive 2-wire elements like rheostats and switches, whereas 3-wire pots and 2-wire control voltage input shall be read unloaded. Changing PUP will cause a full hardware init of the controller, also resetting (auto-)ranging.

INV : *Invert* final output value. Since the output is a 0%..100% range, this simply means the transmitted output value is 100 – controller raw output value. Useful for switches and the occasional pot/rheo which has reverse action (some damper pedals show this behavior). INV can be updated at any time and does not trigger a full hardware init of the controller.

ARAE : *Auto-ranging enable*. When set, a continuous controller remains silent until the input values have reached enough span for stable output, and then the output is normalized to the full 0%...100% range. A bi-stable controller uses a *best guess threshold* derived from the direct input value until auto-ranging has finished. When this bit is cleared and then overwritten with a 1, the auto-ranging limits are reset to initiate a new auto-ranging period. When the bit was set and then is overwritten with a 0, initiating *Manual Ranging Mode*, the limits are changed on the fly and are set to the default value below. ARAE can be updated at any time and does not trigger a full hardware init of the controller.

Range Limits in Manual Ranging Mode:

3-wire Potentiometer	5…95% of full potentiometric range
2-wire Rheostat	5…90% Ref. 10kOhms
control voltage	0…100% Ref. 0V…5V

The default values have been choosen to work with most commercial expressiona damper pedals.

The range limits can be overwritten with the "Ranging" Control Commands.

CONT : Output values are *continuous* (not bi-stable). A bi-stable controller uses the same output value generation like a continuous controller and then applies threshold with hysteresis on it to determine the transmitted output value. CONT can be updated at any time and does not trigger a full hardware init of the controller.

AHS[2:0] : *Auto-hold strength* (0..4) to stabilize and de-noise output values.

- 0 : transmits values as is
- 1..4 : applies increasing levels of auto-holding

The higher the strength the more change of input values (incl. noise) is required to make the controller output eventually follow the input signal. AHS can be updated at any time and does not trigger a full hardware init of the controller.

Auto-Hold is ignored for bi-stable controllers.

2.1 Table of Hardware Source IDs

HWSID[3:0]	Legacy Name	New name
0	**Pedal #1**	**EHC #1**
1	**Pedal #2**	**EHC #2**
2	**Pedal #3**	**EHC #3**
3	**Pedal #4**	**EHC #4**
4	**PitchBender**	PitchBender
5	**AfterTouch**	AfterTouch
6	**Ribbon #1**	Ribbon #1
7	**Ribbon #2**	Ribbon #2
8	-- not used --	**EHC #5**
9	-- not used --	**EHC #6**
10	-- not used --	**EHC #7**
11	-- not used --	**EHC #8**

Note: IDs above 11 (EHC #8) cannot be assigned because those IDs are reserved for special functions in the Audio Engine and Playground. Trying to do so will abort a configuration attempt.

2.2 Table of ADC Channels

CTRLID[2:0]	ADC channel assignment	TRS Jack
0	TRS Jack #1, Tip	#1
1	TRS Jack #1, Ring	
2	TRS Jack #2, Tip	#2
3	TRS Jack #2, Ring	
4	TRS Jack #3, Tip	#3
5	TRS Jack #3, Ring	
6	TRS Jack #4, Tip	#4
7	TRS Jack #4, Ring	

3 EHC_SR: External Hardware Controller Status Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Description	reserved							EER	EES	RAMP	SETLD	RANGD	VALID	RES	PLUGD	INI

EHC_SR can is accessible as a part of the “Get EHC Data” LPC message.

INI : Indicates that a controller is *initialized* and running (active).

PLUGD : Indicates that an active controller is *plugged* in.

RES : Indicates that a controller is running and freshly *reset*, waiting for a plug-in event after which this bit is cleared again.

VALID : Indicates that the controller has sent at least one output value (which can also be read back and is known to be *valid*).

RANGD : Indicates that a controller has finished basic *auto-ranging* stage. When Auto-Ranging is off (ARAE=0), this bit is automatically set.

SETLD : Indicates that a controller raw output value has *settled* within an error band as defined by AHS, the controller then stopping to update the final output value. With Auto-Hold off (AHS=0) this bit is automatically set. Bit is invalid for bi-stable controllers.

RAMP : Indicates that a controller raw output value is *ramping* up or down to continuosly remove, within some time-span, a dynamic offset from the real value. In an active Auto-Hold mode this happens when the output values goes out of settling or has drifted away.

EES : Indicates that the controller state has been saved to the EEPROM. This bit is reset by the “Get EHC Data” command.

EER : Indicates that the controller state has been successfully restored from EEPROM after power-up or reset.

4 Controller Assignment

There are a total of 8 controllers available. When only 2-wire types of controllers are used, all 8 controllers can be assigned and used, simultaneously:

CTRLID[2:0]	Controller Type	TRS Jack
0	2-wire on Tip	#1
1	2-wire on Ring	
2	2-wire on Tip	#2
3	2-wire on Ring	
4	2-wire on Tip	#3
5	2-wire on Ring	
6	2-wire on Tip	#4
7	2-wire on Ring	

Whenever a 2-wire controller is assigned, any previously assigned 3-wire controller that used the associated ADC channel as the secondary ADC (for top contact) is deleted and that secondary ADC input is set to high-impedance state. This is required to correctly read out a 3-wire pot used in a 2-wire rheostat mode.

Controllers that are 3-wire potentiometers can be assigned to either an even Controller ID (indicating wiper contact is on tip) or an odd Controller ID (wiper on ring). For a given TRS jack, only one 3-wire potentiometer can be assigned at a time, hence only 4 controllers are available when all are 3-wire pots:

CTRLID[2:0]	Controller Type	TRS Jack
0 or 1	Pot, Tip or Ring active	#1
2 or 3	Pot, Tip or Ring active	#2
4 or 5	Pot, Tip or Ring active	#3
6 or 7	Pot, Tip or Ring active	#4

Again, whenever a 3-wire potentiometer is assigned, any controllers present on the adjacent channel of the same TRS are deleted.

Note: No attempt is made to restrict assigment of the HWSIDs (Hardware Source IDs). *It is perfectly legal to assign the same HWSID to different controllers working at the same time.* The controller that updates last will dominate. This offers an additional and global way to use multiple controllers (say, a pedal pot and a bench-top pot) controlling the same item without needing to assign multiple HWSIDs to the same Macro Control (which works on a per-preset basis). If controllers update their values at the same time (during one processing time-slice), all changes are transmitted, though, in ascending order by CTRLID, which will result in a very short glitch of the output value for the affected HWSID. If the associated controllers are configured with proper auto-hold this condition is expected to be met only very sporadically, though. The situation is somewhat equivalent to multiple controllers on the same Macro Control.

Of course, 2-wire and 3-wire controllers can be arbitrarily mixed in an EHC setup, the above tables only depict the corner cases. A more general example, showing the extended possibilities, would be:

CTRLID[2:0]	Controller Type	HWSID[3:0]	TRS Jack
0	3-wire, Tip active	6	#1
1	-- auto-disabled --	n/a	
2	2-wire on Tip	3	#2
3	2-wire on Ring	4	
4	-- auto-disabled --	n/a	#3
5	3-wire, Ring active	1	
6	2-wire on Tip	1	#4
7	none	n/a	

That could be a setup for: a 3-wire pot, tip active on TRS #1 mapped to Ribbon#1(6),

a “stereo” switch on TRS#2 mapped tip to EHC#4 and ring to EHC#5,

another 3-wire pot, ring active on TRS#3 mapped to EHC#2,

and finally a single 2-wire rheostat on TRS#4, tip contact, mapped to EHC#2 also.

5 EHC Control Commands (LPC Messages)

5.1 Enable/Disable

The complete runtime processing of the Controllers can be enabled and disabled. This is useful when sending several commands to the EHC module to avoid spurious value updates, for example when changing the ranging.

The LPC message is : `0x0700, 0x0002, 0xFF04, flag` (LPC message "BB_MSG_TYPE_SETTING" with ID "SETTING_ID_ENABLE_EHC").

When *flag* is zero the processing of the controllers is suspended, any other value enables it. Default state (after reset) is enabled.

5.2 Ranging

When auto-ranging is off for a controller, the values to be used for the ranging (scaling) can be set manually. This must always be done *after* controller initialization, with ARAE set to 0. Initialization automatically sets default ranges according to the other settings. It is recommended to disable controller processing until the range values have been written.

Setting the lower range end, the LPC message is: `0x0F00, 0x0002, 0x02nn, value` (LPC message "BB_MSG_TYPE_EHC_CONFIG" with command ID "EHC_COMMAND_SET_RANGE_MIN").

nn is the controller ID CTRLID (0...7). *Value* is an uint16_t representing the lower range end, scaled by the range scale factor (see "Get EHC data").

Setting the upper range end, the LPC message is: `0x0F00, 0x0002, 0x03nn, value` (LPC message "BB_MSG_TYPE_EHC_CONFIG" with command ID "EHC_COMMAND_SET_RANGE_MAX").

If the upper end is not greater than the lower end, the controller will stop outputting data. So, use the INV flag in EHC_CR when you want to revert the direction of movement of a controller.

A typical use case for the ranging commands is to freeze a range that was previously established by auto-ranging, perhaps with some extension of the ends. Since auto-ranging uses values for the actual ranging that are backed off a little from the physical mimimums and maximums, this would otherwise crop, for example, the tops and bottoms of an applied triangular CV (assuming that the CV is moving slow enough that auto-ranging actually does complete, that is). Another example is restricting the range of a pedal with very unstable end stop values which may not be suppressed by by the standard back-off via auto-ranging.

5.3 Forcing Output Update

A controller that is initialized and currently has established a valid output (eg, plugged in, auto-ranging completed, etc) can be forced to output the last value again, sending it to the UI as well as to the Audio Engine (unless output to AE is disabled with the SIL flag in EHC_CR). This useful to implement a Playground feature like "load hardware source values with preset", that is, the hardware sources don't need to be moved to update their values as loaded by a preset) which could be usful. Also, it is nice to be able to retrieve the output value actively, not having to wait for an update event. The LPC message is: `0x0F00, 0x0002, 0x05nn, 0x0000` (LPC message "BB_MSG_TYPE_EHC_CONFIG" with command ID "EHC_COMMAND_FORCE_OUTPUT" and data == 0).

nn is the controller id CTRLID (0...7).

5.4 Reset or Delete Controller

A controller can be reset (resuming to the state it would have after initialization but before a plugin event. Useful especially forcing a new auto-ranging without the need to physically unplug the controller. The LPC message is: `0x0F00, 0x0002, 0x04nn, 0x0000` (LPC message "BB_MSG_TYPE_EHC_CONFIG" with command ID "EHC_COMMAND_RESET_DELETE" and data == 0).

nn is the controller id CTRLID (0...7).

A controller can also be fully deleted (made inactive and all data clear) by sending the LPC message `0x0F00, 0x0002, 0x04nn, 0x0001` (LPC message “BB_MSG_TYPE_EHC_CONFIG” with command ID “EHC_COMMAND_RESET_DELETE” and data != 0).

5.5 Get EHC Data

All relevant data of all controllers can be fetched by issuing the following message to the LPC:

`0x0A00, 0x0001, 0x0002` (LPC message “BB_MSG_TYPE_REQUEST” with request ID “REQUEST_ID_EHC_DATA”).

The response is the actual EHC data message, followed by the standard “Notification” message `0x0800, 0x0002, 0x0002, 0x0001` (LPC Message “BB_MSG_TYPE_NOTIFICATION” with notification ID “NOTIFICATION_ID_EHC_DATA”).

The last word is a flag (!= 0) that the EHC data has been just sent.

The actual EHC data message returned from the LPC is `0x1000, 0x0030, datablock(48 words)` (LPC message “BB_MSG_TYPE_EHC_DATA”).

The EHC *datablock* is structured as 8 sections (one for each controller), each consisting of the following 6 fields (uint16_t):

- EHC_CR : The control register as described above
- EHC_SR : The status register as described above
- Last Value : Last output value (0...16000), only valid if VAL bit is set in EHC_SR.
- Range Minimum : currently used lower end value for ranging
- Range Maximum : currently used lower end value for ranging
- Range Scale Factor : scale factor for the range values

The range-related fields are only valid for initialized controllers (EHC_SR:INI = 1).

The scale factor varies depending on the type of controller and it shall be used to display the range values in percent (%).

Note: Range values above 100% may appear for 2-wire rheostat type of controllers where scale factor is based on a nominal 10kOhms resistance. For 3-wire potentiometric readout the scaled values usually never exceed 100% (as the scale factor is the value corresponding to a wiper-to-top voltage ratio of 1.0, which can only be exceeded by noise contribution and other dynamic ADC errors), and for a CV they cannot exceed 100% by definition because the scale factor is the full ADC range.

After “Get EHC Data”, the EES bit in the status register EHC_SR will be cleared, for all controllers.

5.6 Save EHC Data to EEPROM

While the EEPROM is automatically updated whenever the state of a controller changes, a forced save to EEPROM with all current controller data is possible. The LPC message is `0x0A00, 0x0001, 0x0005` (LPC message “BB_MSG_TYPE_REQUEST” with request ID “REQUEST_ID_EHC_EEPROMSAVE”).

The response is a “Notification” message `0x0800, 0x0002, 0x0005, flag` (LPC Message “BB_MSG_TYPE_NOTIFICATION” with notification ID “NOTIFICATION_ID_EHC_EEPROMSAVE”).

The last word is a flag (!= 0) that the EHC data writing process was succesfully started. It may fail though (returning 0), when the EEPROM is currently busy (or another more severe fault condition is present). Then a retry may be initiated.

6 Auto-Save/Auto-Restore to/from EEPROM

From time to time there is a check if a controller's state has changed, and if so the state is written to a non-volatile memory (EEPROM). The saved state is comprised of the configuration and the current values needed for (auto-)ranging.

At boot time (at power-up or after a reset), if a valid state is found then the controller is initialized with it. With no valid state found (corrupt data in the EEPROM) the controller is cleared and initialization will be done by the playground when it starts. The PG may overwite the controllers with another setup, of course.

Auto-Restore is very convenient notably with the auto-ranging feature because the auto-ranging is not required to start from a fully cleared state the next time the C15 is powererd up again and nothing had been changed with regard to the physical hookup of the external hardware controllers, say, in a typical studio situation.

7 Compatibility

The legacy “Pedal Configuration” Control Commands issued to the LPC by the current Playground versions are supported (LPC Messages “BB_MSG_TYPE_PARAMETER” with parameter IDs “SETTING_ID_PEDAL_1_TYPE” through “SETTING_ID_PEDAL_4_TYPE”).

They are implemented as fixed presets of configuration register data that are passed to EHC_CR.

The legacy controls can only assign controllers to legacy HWSIDs in the range of EHC#1(0) to EHC#4(3) (aka Pedal#1 to Pedal#4), and the mapping of CTRLID's to HWSID's is such that the HWSID corresponds directly to the TRS Jack (TRS#1 to TRS#4) used.

Configuration of the legacy “Pedal Types” (“n” denoting the HWSID):

Pedal, Tip Active:

HWSID[3:0]	SIL	CTRLID[2:0]	POT	PUP	INV	ARAE	CONT	AHS[2:0]
n	0	2*n	1	0	0	1	1	2

Pedal, Ring Active:

HWSID[3:0]	SIL	CTRLID[2:0]	POT	PUP	INV	ARAE	CONT	AHS[2:0]
n	0	2*n + 1	1	0	0	1	1	2

Switch, Closing:

HWSID[3:0]	SIL	CTRLID[2:0]	POT	PUP	INV	ARAE	CONT	AHS[2:0]
n	0	2*n	0	1	0	1	0	0

Switch, Opening:

HWSID[3:0]	SIL	CTRLID[2:0]	POT	PUP	INV	ARAE	CONT	AHS[2:0]
n	0	2*n	0	1	1	1	0	0

Note:

All legacy types are set up with Auto-Ranging enabled. Together with the Auto-Restore this means that the Auto-Ranging limit values are not reset at power-up (as long as the same type is used) and no manual cycling of the pedals for Auto-Ranging is required. In order to reset the Auto-Ranging two methods can be used:

– Unplug and replug the pedal while the C15 is running. This is the recommended method because it is guaranteed to work at any rate.

– Temporarily select a different type of controller in the setup. If one of “pedal” types is currently active, select a “switch” type, and vice versa. It is not sufficient, though, to switch from one “switch” type to the other and back because the difference is only the ouput inversion which is changed on-the-fly, without a reset of the controller.