Charger protocol

Chan	From CCU to Charger	Chan	From Charger to CCU	Meaning
Probing and connection:				
		360	ACS,hello,CHARGER	Probe for a control system
360	ACS,welcome,*device*,*information[,...]*			Controller reports its existence. For a CCU, *device*=ccu Your device *should* report something different but the current charger will ignore anything except ccu, so report that.
		360	ACS,interface,CHARGER	Request an interface
360	ACS,interface,*channel*			Provide the charger with a channel on which to communicate. Use a large random negative number.
360	ACS,refuse,*reason*			The CCU may refuse the connection.
The charger and CCU can now communicate back and forth on the specified *channel*, which no other devices should be sharing. The following messages are sent by the CCU immediately after the connection is established, and may be sent again at any time later. Any message from the charger may optionally contain the key of the charger's operator inserted after "ACS," followed by a caret (^), as in: ACS,*key*^chargersummary:				
channel	ACS,powertype:EL			Tells charger this unit uses electrical power. If this is not received, or has a different value, charger won't charge it.
channel	ACS,maxcharge:*maxcharge*			*maxcharge*: Number of ticks of charge the current power source can hold.
The charger requests the following information immediately after the connection is established, and may be requested at other times.				
		channel	ACS,chargersummary:	Request information about the unit's current charge state
channel	ACS,chargersummary:*power,chargepct,tickstofull*			Reply to chargersummary: *power*: 0 if shut down, 1 if active *chargepct*: 0-100; 0=dead,100=fully charged *tickstofull*: Number of ticks of charge required to fully charge
The following messages constitute the normal charging sequence:				
		channel	ACS,charging:1	Alerts the CCU that charging has begun (so it can indicate this on the HUD). No reply expected.
		channel	ACS,chargeseconds:*n*	Provides *n* ticks of charge. (*n* is typically 150 or 600, but can in principle by any number.) How the power supply handles overcharging is up to it. CCU should send chargeticks: (below)
		channel	ACS,setcharge+:*n*	Synonym for chargeseconds. Please accept either.
channel	ACS,chargeticks:*ticks*			CCU reports current charge level to charger when a significant

Chan	From CCU to Charger	Chan	From Charger to CCU	Meaning
				change occurs, including any increase.
		channel	ACS,charging:0	Alerts the CCU that charging is over.
The following messages may be sent by the CCU at any time (while connected):				
channel	ACS,dump:*key*,*string*			If *string* includes the name of a script in the charger, that script will send debugging information to *key*.
channel	ACS,unitdisconnect:*n*			If *n*=1, unit is allowed to disconnect itself from the charger at will. If *n*=0, unit cannot disconnect itself while charging.
channel	ACS,power:*n*			Sent if unit's power turns on (*n*=1) or off (*n*=0).
channel	ACS,stopcharge:			Signals the charger to stop charging.
		channel	ACS,travel:*n*	Tells the CCU to prevent (*n*=0) or allow (*n*=1) the unit from moving from its current spot
The following sequence disconnects the charger from the CCU.				
		channel	ACS,disconnect:	Tells the CCU to disconnect. (We expect goodbye in response)
channel	ACS,goodbye:			The CCU is disconnecting. (Need not be provoked by disconnect: if the CCU chooses to initiate disconnection.)
360	ACS,goodbye:			Same as above. Sent if CCU resets, losing channel information.

S