External ICD

REF: CP030-Doc2-External_ICD

01 de April de 2019

Versión: V2.0

Author: JLC

Approved by: JLC

Editado por:

Cerro Electronic Design

Avda. Cerro del Águila Nº9

28703 San Sebastián de los Reyes

Madrid, Spain.

Distribution List

Table 1: Distribution List

Company	Name	Position	No. Of copies
	Public distribution		

ÍNDICE DE CONTENIDOS

Document Revision History

Table 2: Document Revision History

Version	Issue date	Brief Description of the change
V1.0	25-Sept-2017	Primera versión del documento
V1.1	03-Octu-2017	- Se eliminan los mensajes TRIGGER AVERAGE y AVERAGE FRAME - Se crean los mensajes SET SAMPLES y VALUE SAMPLES
V1.2	20-Octu-2017	- Se añaden nuevos códigos de error en el apartado 0 Note that with this message you can do RESET of individual boards. - ERROR - Se añaden ejemplos de comandos en el apartado 0, The payload of the message contains the two identifier bytes of the command that has been taken as erroneous and its X, Y, Z coordinates. - - Se añade el comando ACK_HARDWARE (apartado 3.2.12) y en este mismo apartado se añade la polaridad de la señal hardware que produce el disparo
V1.3	01-Feb-2018	- Corrección del Baudrate en **¡Error! No se encuentra el origen de la referencia.** - Aclaración sobre el mensaje Start, apartado 3.2.1
V2.0	27-03-2019	-Doc. Translation into English - Adaptation to V2.0 of the board

LISTA DE FIGURAS

LISTA DE TABLAS

1 Introduction

This document describes the external control interface of the PhotoArray board, an array of photodiodes in a matrix of 9x6 pitch 9mm designed by Cerro.

PhotoArray consists of an array of photodiodes and all the necessary elements to measure the current of each of them, digitize it and send it through a serial port.

Following picture shows a graphical view of the incident light to the different photo diodes (or pixels) on the board:

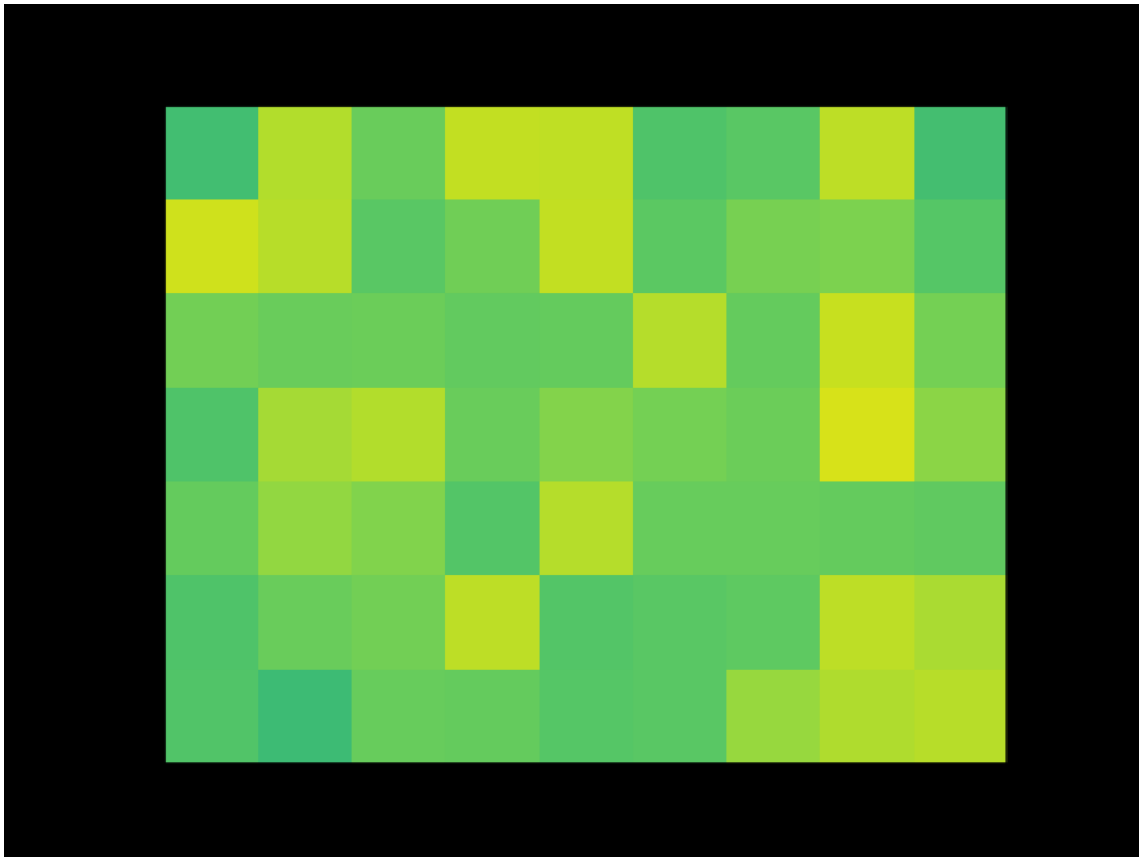

Figure 1: Graphical representation on incident light

PhotoArray is based on a Cypress PSOC system that manages the readings of the photodiodes. After reading and processing the data, they are sent by request, to a master system through serial communication port as described in this document.

1.1 Acronyms

ADC, Analog Digital Converter

BGA, Ball Grid Array

DAC, Digital Analog Converter

ICD, Interface Control Document

GPIO, General Purpose Input Outputs

NO, Normally Open

NC, Normally Close

PCB, Printed Circuit Board

OC, Open Circuit

PSOC, Programmable System on Chip

RP, Master

SC, Short Circuit

SMD, Surface Mount Device

SMT, Surface Mount Technology

TBD, To Be defined

TBC, To Be Confirmed

1.2 Document property

The information contained in this document belongs to the intellectual property of Cerro Electronic Design S.L. This information is strictly confidential and cannot be copied or distributed by third parties without the written permission of the company itself

2 HW Interface

2.1 Introduction

Communication with the PhotoArray board take place, at the physical level, through an RS-485 serial port with the following features

Table 3: RS485 Configuration

Parameter	Value
Baud Rate	57600
Data Bits	8
Stop Bits	1
Parity	No
Flux control	No
Configuration	Half duplex

For communication with a PC you can use the free software "Realterm" and a USB-RS485 converter such as the reference USB-RS485-WE-1800-BT, Farnell Code: 1740357.

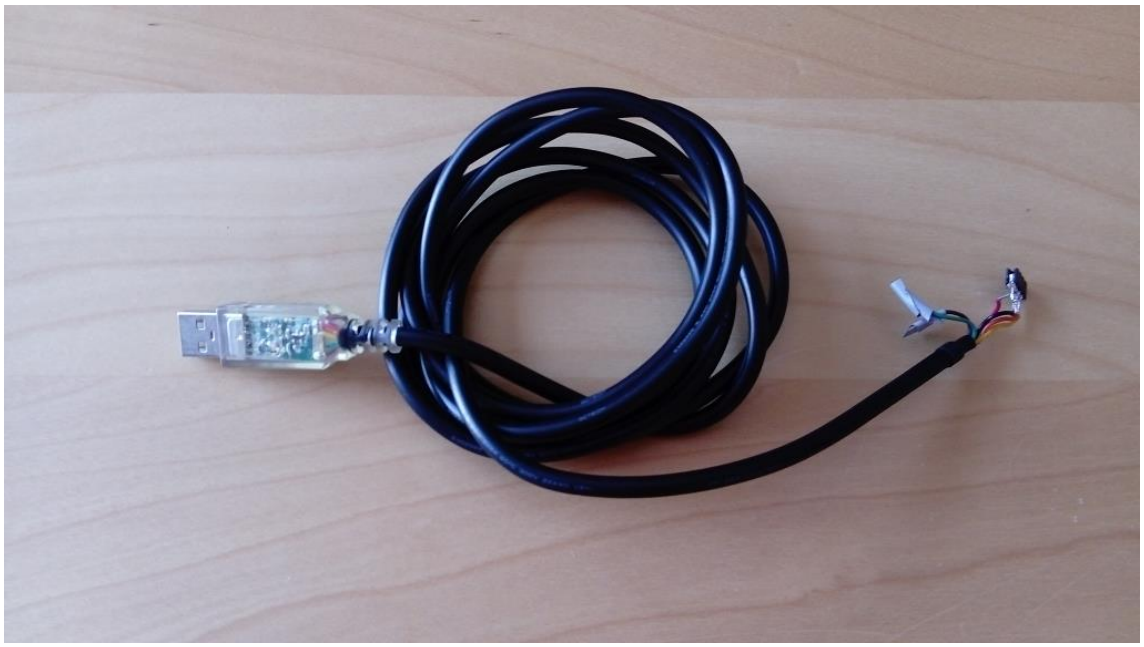

Figure 2: USB to RS485 converter

Up to 16 PhotoArray boards can be connected on the same bus. The address of each board can be configured using microswitch SW1.

2.1 Board ID

ID of each board can be configured with the SW1 microswitch. Next table specify the microswitch position for each address ID.

Table 4. ID configuration

ID	ID0	ID1	ID2	ID3
0	off	off	off	off
1	on	off	off	off
2	off	on	off	off
…	…	…	…	..
15	on	on	on	on

2.2 RS485 Communication

The following image shows schematically the communications between the PhotoArray system and a Master system, which for reasons of understanding we can say that it is a RaspberryPi.

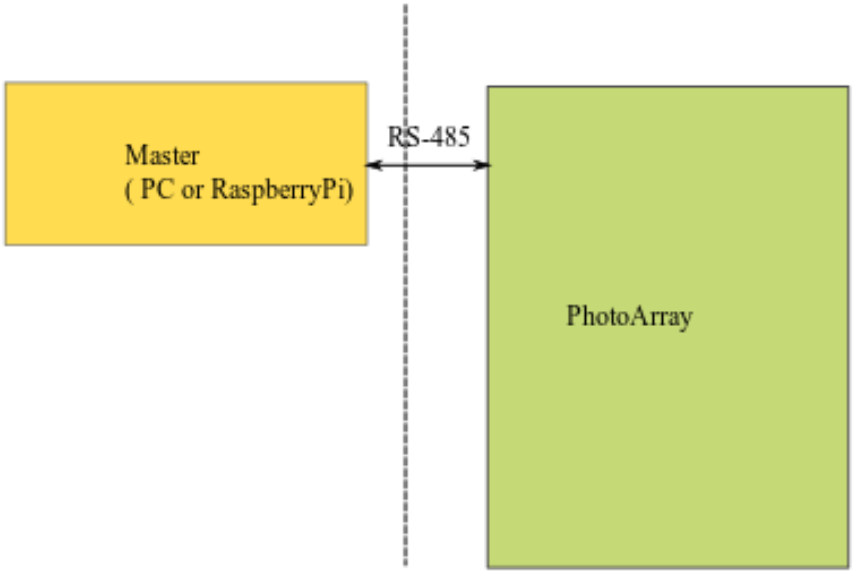

Figure 3: Comunicaciones Master/Photoarray

2.3 Scalability

To give PhotoArray greater versatility and be able to address projects that require the use of more photodiodes, PhotoArray will implement a scalability system. The following figure shows the concept.

Port 0 and 1 are just two identical connectors with the same pinout, so we can pull a cable from one board to another. That is, the data to be transmitted / received will be available in the two connectors.

Each photodiode in the full system is identified by its coordinates (X, Y, Z). X and Y refer to the coordinates of the diode within its board and Z refers to the ID board within the system.

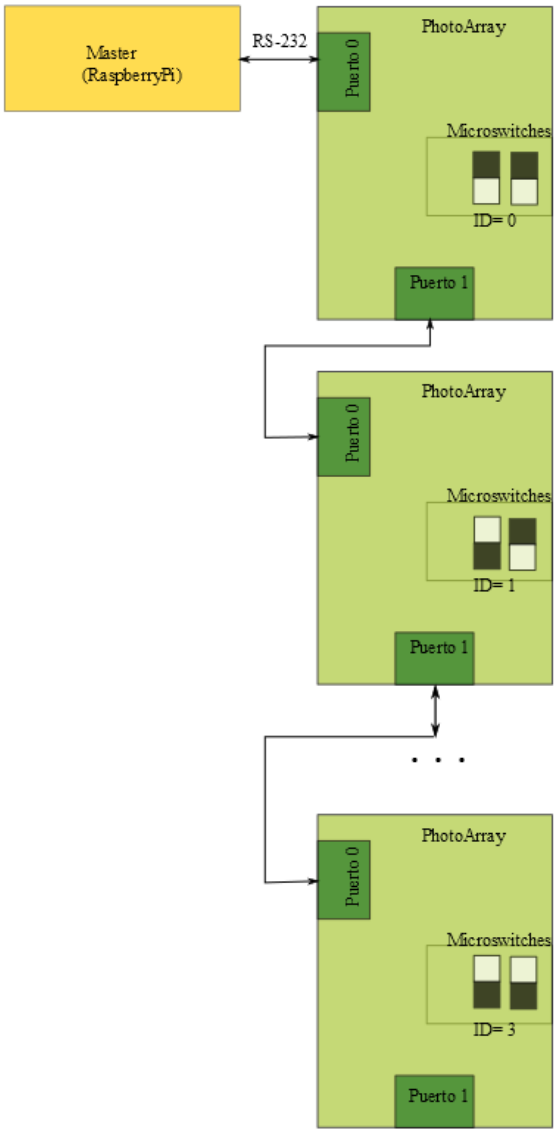

Figure 4: Several board in the same bus

3 Software Protocol

3.1.1 Protocol description

During the normal operation of the system, an exchange of messages between Master and PhotoArray will take place.

Both systems will send and receive different types of messages depending on the current state of the measurement process and depending on the options and actions carried out by the user, or other events that may occur.

The messages exchanged between each board can be grouped into 3 groups:

• Photodiode **measurement request messages**. the information referring to the photodiode current. Each sensor, that is each photodiode, in the message will be identified.

• **Start messages**: initial information exchange messages (identification of boards on the bus).

• **Error message** (error). Errors messages sent by boards when any error produced

Below are some diagrams to show the typical message exchange for each of the previous types.

The **measurement request messages** (GET-x-y-z) are followed by the required value.

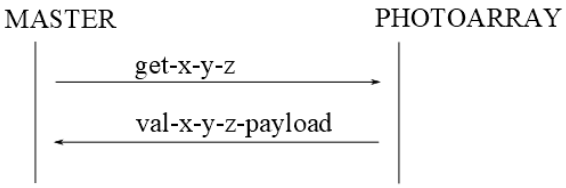

Figure 5 : Get messages diagram

Start Messages

At startup, the Master system sends a start message (INIT) to identify the PhotoArray boards connected to the system. Each PhotoArray will send a message with its identifier to indicate to the system how many cards are available (Z coordinate).

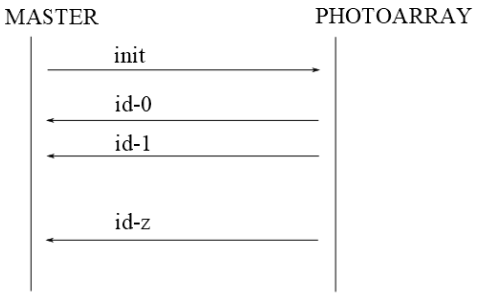

Figure 6: Start messages diagram

3.1.2 Messages format

The messages have a size of 11 bytes, except when a full frame is sent as will be seen later. The commands will consist of the following fields:

- Start Byte: message start value coded with the value 0x55
- Command: identifier of the message type
- X coordinate: identifier of the photodiode column
- Y coordinate: identifier of the photodiode row
- Z coordinate or ID: plate identifier (ID) defined by hardware microswitches
- Payload: value attached to the message (see types in Table 5). There are cases that these bytes will not be necessary, they will be filled with 0. Their size will be 4 bytes for all the commands except for "Full Frame" which will be 252 bytes.
- End Bytes: message end bytes encoded with 0x0D 0x0A.

Tabla 5 : Messages format

Start Byte	Command Byte	COORDs (X-Y)	COORDs (Z)	Payload	End Bytes
1 byte	2 bytes	1 byte	1 byte	4 bytes	2 bytes

3.1.3 Command Types

Following table shows all the commands available

Tabla 6. Message Types

COMANDO	Payload Type	Origen	Destine	Respuesta a
START(1)	FW Version	Master	PhotoArray	-
INIT	Fill with zeros	Master	PhotoArray	-
ID Z	Fill with zeros	PhotoArray	Master	Init
GET CURRENT X-Y-Z	Fill with zeros	Master	PhotoArray	-
VAL CURRENT X-Y-Z	Uint32	PhotoArray	Master	GET CURRENT X-Y-Z
SET SAMPLE-Z	Value of Samples	Master	PhotoArray	-
VALUE SAMPLE-Z	Uint32	PhotoArray	Master	SET SAMPLE-Z
GET FRAME	Fill with zeros	Master	PhotoArray	
FULL FRAME	252 Bytes	PhotoArray	Master	GET FRAME
TRIGGER SOFTWARE	Fill with zeros	Master	PhotoArray	
ACKNOWLEDGE SOFTWARE	Fill with zeros	PhotoArray	Master	TRIGGER SOFTWARE
ACKNOWLEDGE HARDWARE	Fill with zeros	PhotoArray	Master	HARDWARE SOFTWARE
GET TEMP Z	Fill with zeros	Master	PhotoArray	-
VAL TEMP Z	Int16	PhotoArray	Master	GET TEMP Z
RESET	Fill with zeros	Master	PhotoArray	-
ERROR	Error Type	PhotoArray	Master	Any erroneous message

(1) See chapter 3.2.1

3.2 Message Description

3.2.1 START Message

Important Note: This message has not been implemented. It has been replaced by an ASCII text similar to this "Start Version Vx.x".

The message is sent first by all the UART (communications and Debug).

In order to avoid that the different cards send your START message at the same time, and a "data shock on the bus" occurs, a delay will be assigned depending on ID board, so for example ID = 0, you will send your message as soon as possible, ID = 1 will send it with a delay of 200ms, ID = 2 with 400ms and ID = 3 with 600mS and so on

The description that follows is not implemented but is left in case it is necessary to implement it.

The START message is sent by the PhotoArray when the board is started or after a reset process performed internally by the Watchdog. In the PAYLOAD, the version of FW that the board has is encoded (Ex: "01.3"). The XY coordinate fields are kept at zero and Z coordinate the ID of the board

Start Byte	Command Byte	COORDs (X-Y)	COORDs (Z)	Payload	End Bytes
0x55	0x53 x54	0x00	Id	FW	0x0D 0x0A

Tabla 7. START Message

3.2.2 INIT message

The INIT message is sent by the master system and indicates that the system is going to start and requests the connected PhotoArray cards to be identified, that is, master asks all the cards to respond with an ID message. It is waiting for an identification message of each plate.

Start Byte	Command Byte	X-Y	Z	Payload	End Bytes
0x55	0x49 x4E	0x00	0x00	0x00000000	0x0D 0x0A

Tabla 8. INIT Message

3.2.3 ID Message

The identification message of each PhotoArray connected to the bus. It sends in its Z coordinate the value configured in its micro-switches (digital value).

Start Byte	Command Byte	X-Y	Z	Payload	End Bytes
0x55	0x49 0x44	0x00	ID	0x00000000	0x0D 0x0A

Tabla 9. ID Message

In order to avoid that the different cards send their ID at the same time, and a "data shock on the bus" occurs, they will be assigned a delay depending on their ID, so for example ID = 0, will send its message as soon as possible, ID = 1 will send it with a delay of 200ms, ID = 2 with 400ms, ID = 3 with 600ms and so on.

For example, a message ID = 03 would be received in the master system as follows:

0x55 0x49 0x44 0x00 **0x03** 0x00 0x00 0x00 0x00 0x0D 0X0A

3.2.4 SET SAMPLES Message

It establishes the number of samples taken by the ADC to measure the current of a photodiode. Thus, if the SET SAMPLE = 1 value, the ADC only samples the current of the photodiode once and if SET_SAMPLE = 10, it will take 10 measurements and will average those 10 measurements to give the value of the current.

This value will be maintained for all measurements from that moment until it is modified or the card is reset.

- XY Coordinate: 0x00 value
- Z coordinate: board identifier defined by microswitches (value uint8).
- Payload, value of the number of samples to be taken (0xSS)

Start Byte	Command Byte	X-Y	Z	Payload	End Bytes
0x55	0x53 0x53	0x00	Uint8	0xSS 0x000000	0x0D 0x0A

Tabla 10. Set Sample message

The default value at startup of all cards is 1.

3.2.5 VALUE SAMPLES Message

"Acknowledge" message sent by the card when the number of samples is configured with the SET SAMPLE command.

Start Byte	Command Byte	XY	Z	Payload	End Bytes
0x55	0x56 0x53	0x00	ID	0xSS 0x000000	0x0D 0x0A

Tabla 11. Value Samples Message

In the Payload, the least significant bytes are sent first, for example, a message that returns the ID = 1 board, when the samples to be taken of 10 (0x0000000A) have been configured, would be received in the master system as follows:

0x55 0x56 0x53 0x00 0x01 **0x0A 0x00 0x00 0x00** 0x0D 0X0A

3.2.6 GET CURRENT Message

Message to get the current values in each photodiode (X, Y, Z).

- XY coordinate: identifier of the photodiode on the board (value uint8).
  Being the most significant bits those corresponding to the X coordinate (columns) and the 4 least significant ones corresponding to the Y coordinate (rows).
- Z coordinate: plate identifier defined by microswitches (value uint8).

Start Byte	Command Byte	X-Y	Z	Payload	End Bytes
0x55	0x47 0x43	Uint8	Uint8	0x00000000	0x0D 0x0A

Tabla 12. Get Current Message

3.2.7 VAL CURRENT message

Measured digital value of the photodiode in the X-Y-Z coordinates sent from the acquisition card. The measured value will be sent as a 32-bit unsigned integer

Start Byte	Command Byte	XY	Z	Payload	End Bytes
0x55	0x56 0x43	Uint8	Uint8	Unsigned Int 32	0x0D 0x0A

Tabla 13. Value Current Message

In the Payload, the least significant Bytes are sent first, for example, a message that returns the photodiode current in position (3,2,1), whose current in hexadecimal value is 0x12345678, would be received in the master system as follows:

0x55 0x56 0x43 0x32 0x01 **0x78 0x56 0x34 0x12** 0x0D 0X0A

3.2.8 GET FRAME Message

With this message, the last available values of the current of all the photodiodes on a card are requested. It does not take a new frame.

- Z coordinate: board identifier defined by microswitches (value uint8).

Start Byte	Command Byte	X-Y	Z	Payload	End Bytes
0x55	0x47 0x46	0x00	Uint8	0x00000000	0x0D 0x0A

Tabla 14. Get Frame message

3.2.9 FULL FRAME message

This message is the response of the board to the GET FRAME message. Send the current values of the card (it does not take a new frame).

Start Byte	Command Byte	X-Y	Z	Payload	End Bytes
0x55	0x46 0x46	0x00	Uint8	252 Bytes	0x0D 0x0A

Tabla 15. Full Frame Message

Payload being as follows: the first 4 bytes transmitted correspond to the photodiode (0,0), the seconds to (0,1) and so on until the end of the photodiode data (6,8).

First data	(0,0)	(1,0)	(2,0)	(3,0)	(4,0)	(5,0)	(6,0)	(7,0)	(8,0)
	Int32	Int32	Int32	Int32	Int32	Int32	Int32	Int32	Int32

(0,1)	(1,1)	(2,1)	(3,1)	(4,1)	(5,1)	(6,1)	(7,1)	(8,1)
Int32	Int32	Int32	Int32	Int32	Int32	Int32	Int32	Int32

(0,2)	(1,2)	(2,2)	(3,2)	(4,2)	(5,2)	(6,2)	(7,2)	(8,2)
Int32	Int32	Int32	Int32	Int32	Int32	Int32	Int32	Int32

…

…

(0,6)	(1,6)	(2,6)	(3,6)	(4,6)	(5,6)	(6,6)	(7,6)	(8,6)	last data
Int32	Int32	Int32	Int32	Int32	Int32	Int32	Int32	Int32	

Therefore, the size of the Payload in this command is: 4x9x7 = 252 bytes
The full size of the message will be: 7 + 252 = 259 bytes.

3.2.10 TRIGGER SOFTWARE Message

This message tells the board to take a new full frame.

Start Byte	Command Byte	X-Y	Z	Payload	End Bytes
0x55	0x54 0x53	0x00	ID	0x00000000	0x0D 0x0A

Tabla 16. Trigger Software Message

3.2.11 ACKNOWLEDGE SOFTWARE Message

This message indicates that a complete frame has been taken in response to the TRIGGER SOFTWARE message and is sent once the operation has been completed to indicate that the frame is available. If you wanted the frame taken, you would have to request it using the GET_FRAME command.

Start Byte	Command Byte	X-Y	Z	Payload	End Bytes
0x55	0x41 0x53	0x00	ID	0x00000000	0x0D 0x0A

Tabla 17. Acknowledge Software Message

3.2.12 Mensaje ACKNOWLEDGE HARDWARE

This message indicates that a complete frame has been taken in response to a TRIGGER HARDWARE and is sent once the operation has been completed, to indicate that the frame is available. If you wanted the frame taken, you would have to request it using the GET_FRAME command.

Start Byte	Command Byte	X-Y	Z	Payload	End Bytes
0x55	0x41 0x48	0x00	ID	0x00000000	0x0D 0x0A

Tabla 18. Acknowledge Hardware Message

In order to avoid that the different cards send their ACK_HARDWARE at the same time, and a "data shock on the bus" occurs, they will be assigned a delay depending on their ID, so for example ID = 0, it will send their message as soon as possible. ID = 1 will send it with a delay of 200ms, ID = 2 with 400ms, ID = 3 with 600mS and so on.

The trigger signal is set to be active at low level, that is, in its idle state it will have a "High" logic level. The falling edge is detected, so the pulse width does not matter.

3.2.13 GET TEMPERATURE Message

Each card has a single temperature sensor and this is defined by the ID value of the board. To request the temperature of the sensor, the following message will be used:

- Z coordinate: ID board identifier defined by microswitches (value uint8).

Start Byte	Command Byte	XY	Z	Payload	End Bytes
0x55	0x47 0x54	0x00	ID	0x00000000	0x0D 0x0A

Tabla 19. Get Temperature Message

3.2.14 VALUE TEMPERATURE Message

Temperature data sent from the PhotoArray card. The temperature value will be sent with a variable type int16.

Start Byte	Command Byte	XY	Z	Payload	End Bytes
0x55	0x56 0x54	0x00	Uint8	Temperature value/100	0x0D 0x0A

Tabla 20. Value Temperature message

Temperature value: int16, temperature in 1/100 ºC of the sensor.

3.2.15 Mensaje RESET Message

The RESET message is sent by the control system (master) in order to perform a reset of the acquisition system (Photoarray boards)

Start Byte	Command Byte	XY	Z	Payload	End Bytes
0x55	0x52 0x53	0x00	ID	0x00000000	0x0D 0x0A

Tabla 21. Reset Message

Note that with this message you can do RESET of individual boards.

3.2.16 ERROR Message

The message ERROR indicates that the received data is not correct, depending on the error code it will be possible to detect at which point the fault has occurred.

Start Byte	Command Byte	0x00	Error Code	Payload	End Bytes
0x55	0x45 x52	0x00	Byte Error	MSG_ID COOD_Y COOD_X COORD_Z	0x0D 0x0A

Tabla 22. ERROR Message

The error codes can be:

Error Code	Description
0x30	UART identifier corrupted, due to possible wrong access to memory. Reset the system.
0x31	Badly formed message, the corresponding bytes between the START and END bytes have

	not been received.
0x32	"Command Byte" field of the message is wrong (does not exist…).
0x33	Incorrect XY coordinate
0x34	Error in temperature sensor. Try again the measurement and if it persists it is possible that it is an error of HW
0x35	SET_SAMPLES (number of ADC samples) cannot be equal to zero or greater than 255. Number of Samples not modified.

Tabla 23. Error codes

The payload of the message contains the two identifier bytes of the command that has been taken as erroneous and its X, Y, Z coordinates.

4 COMMANDS EXAMPLES

This chapter shows how to send commands through a serial port of the PC so that the Master system is simulated. It is the simplest way to test the functionality of the card.

For communication with a PC you can use the free software "Realterm" and a USB-RS485 converter such as the reference USB-RS485-WE-1800-BT, Farnell Code: 1740357.

4.1 START message

Message at start-up of the board

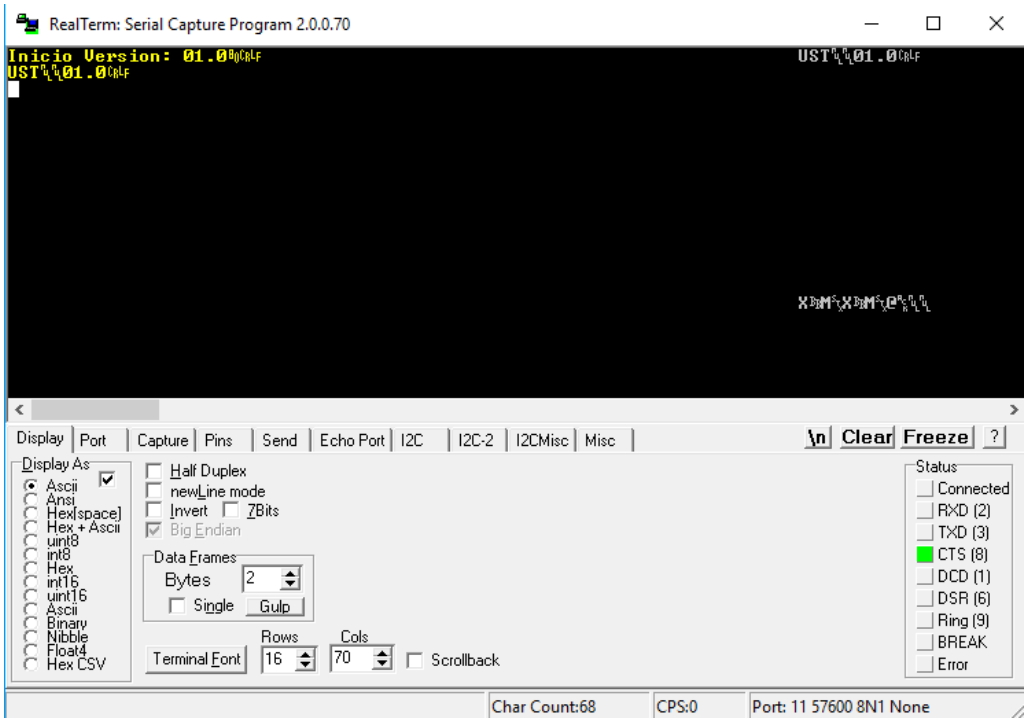

Figure 7 : START message in ASCII

The same message in Hex format:

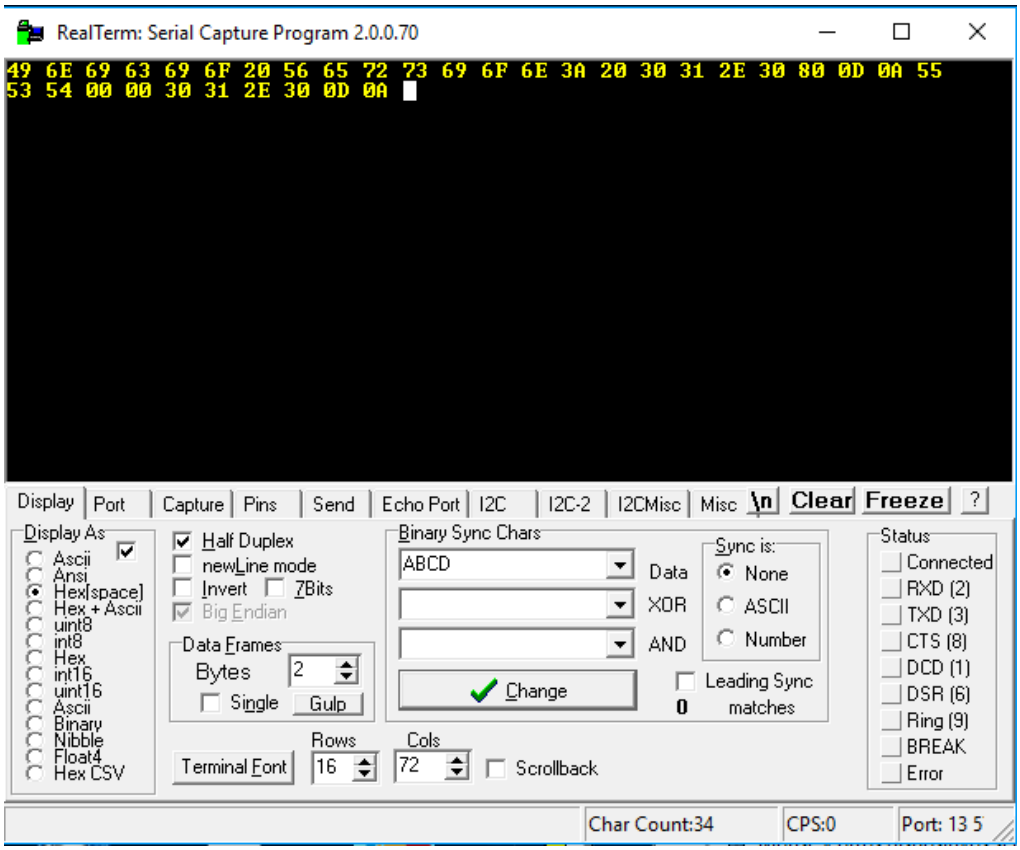

Figure 8: START message in hexadecimal format

4.2 INIT & ID Commands

The following image shows the sending of an INIT message and the responses of boards with ID = 0 and ID = 1.

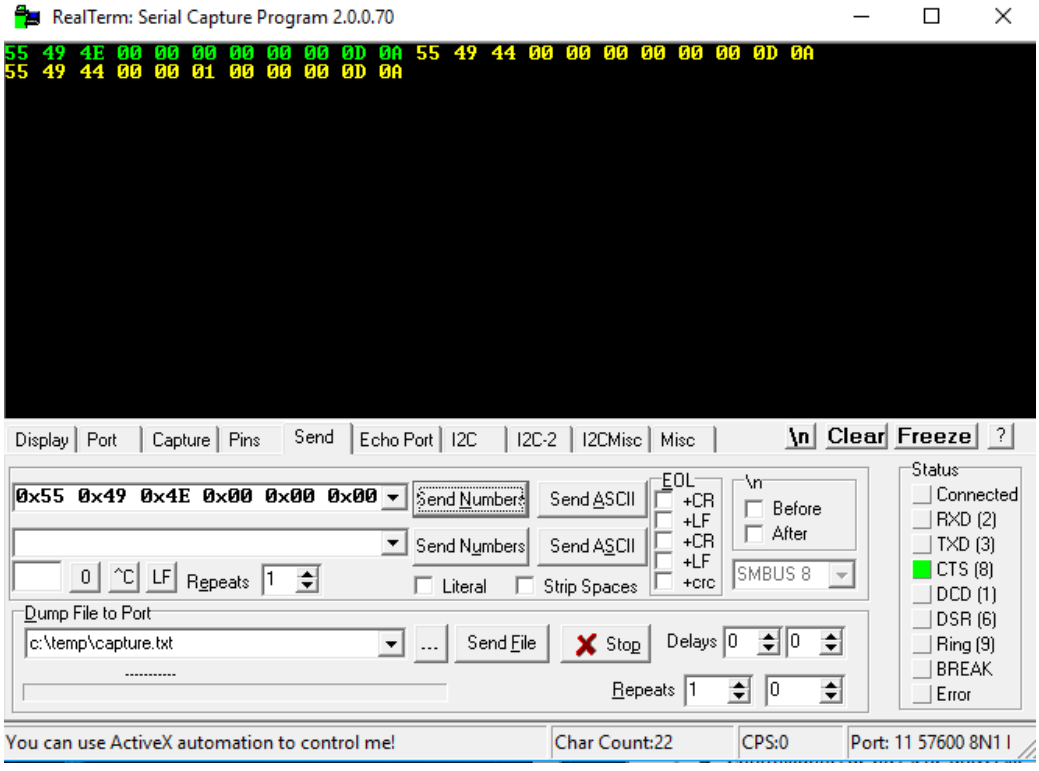

Figure 9: INIT & ID messages for boards ID=0 and ID=1

4.3 SET_SAMPLE Command

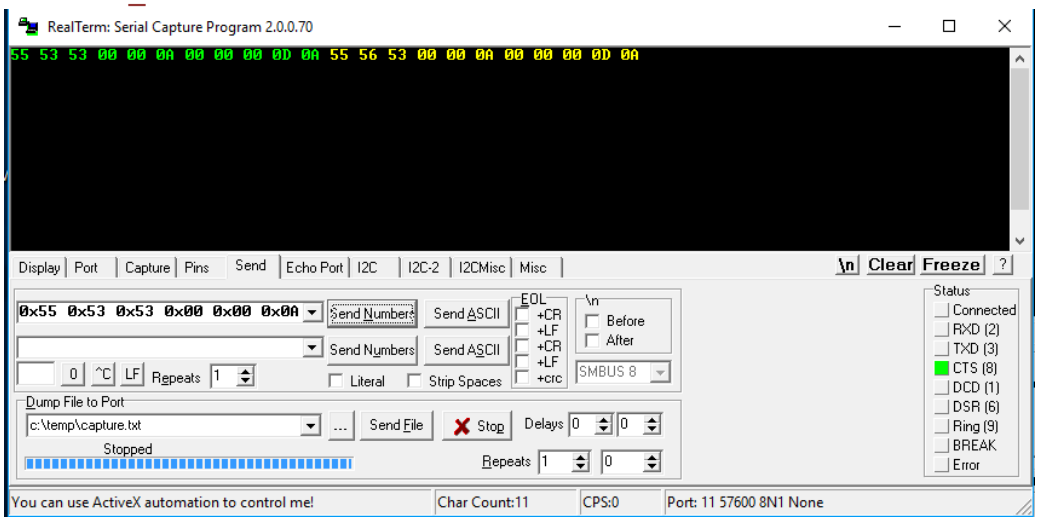

Figure 10: SET_SAMPLES Command

4.4 GET_CURRENT Command

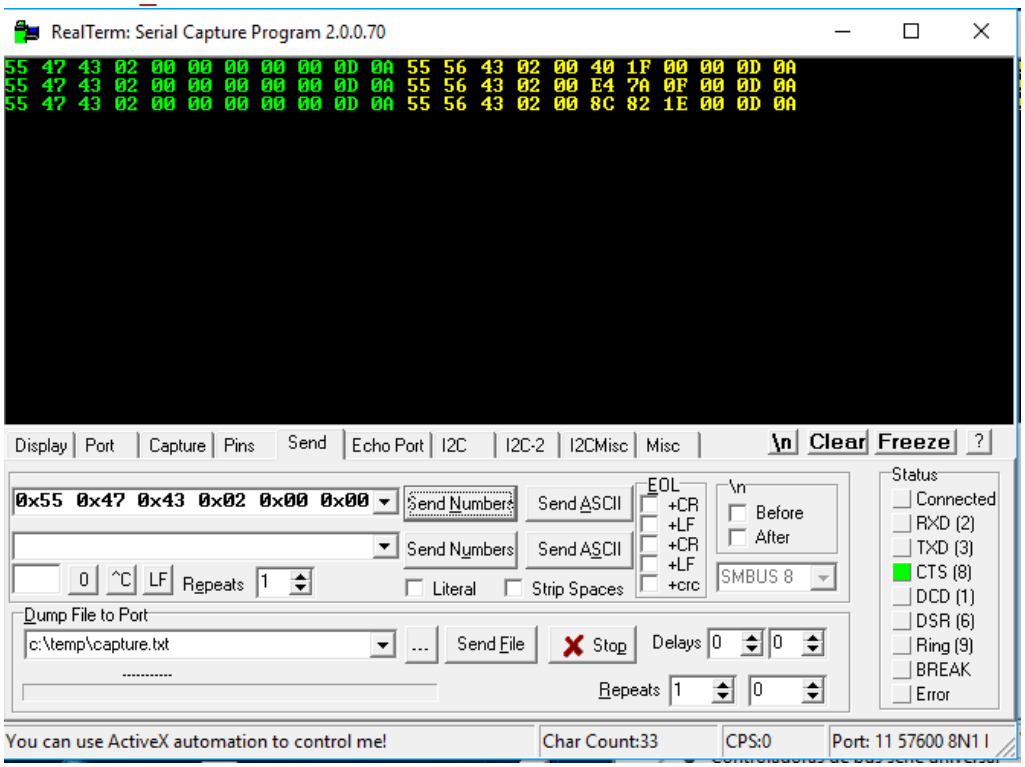

Figure 11:Three GET_CURRENT messages and the answers

In the previous image you can see three examples of the calculation of the photodiode voltage (0,2) for different values, the first of them for 0 Volts, the second 1 Volt and the third of 2V.

4.5 TRIGGER_SOFTWARE Command

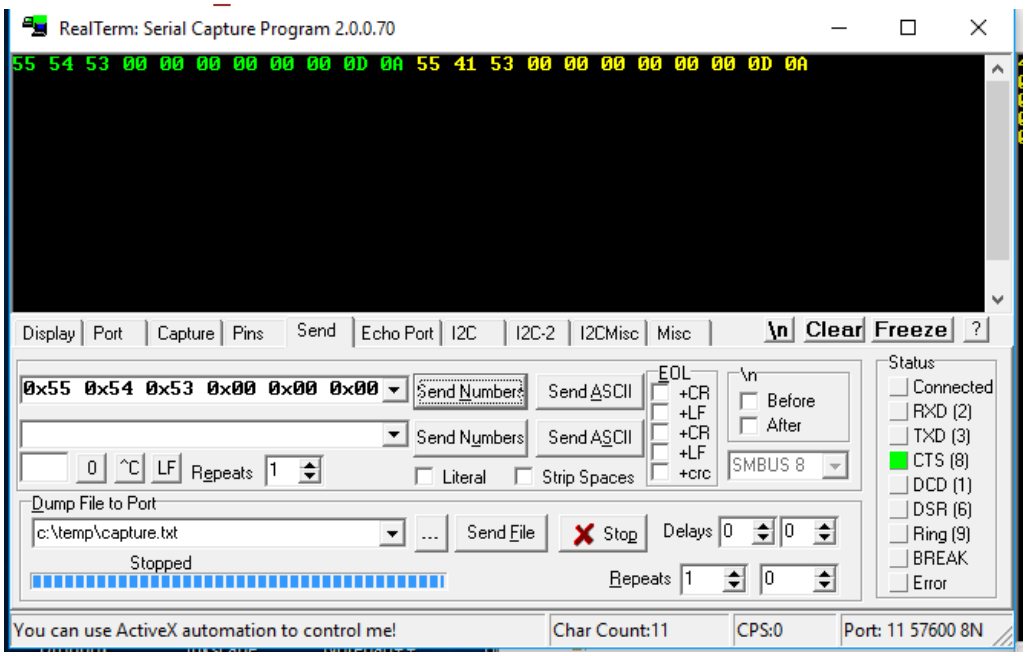

Figure 12 : TRIGGER_SOFTWARE command

4.6 GET_CURRENT y GET FRAME Commands

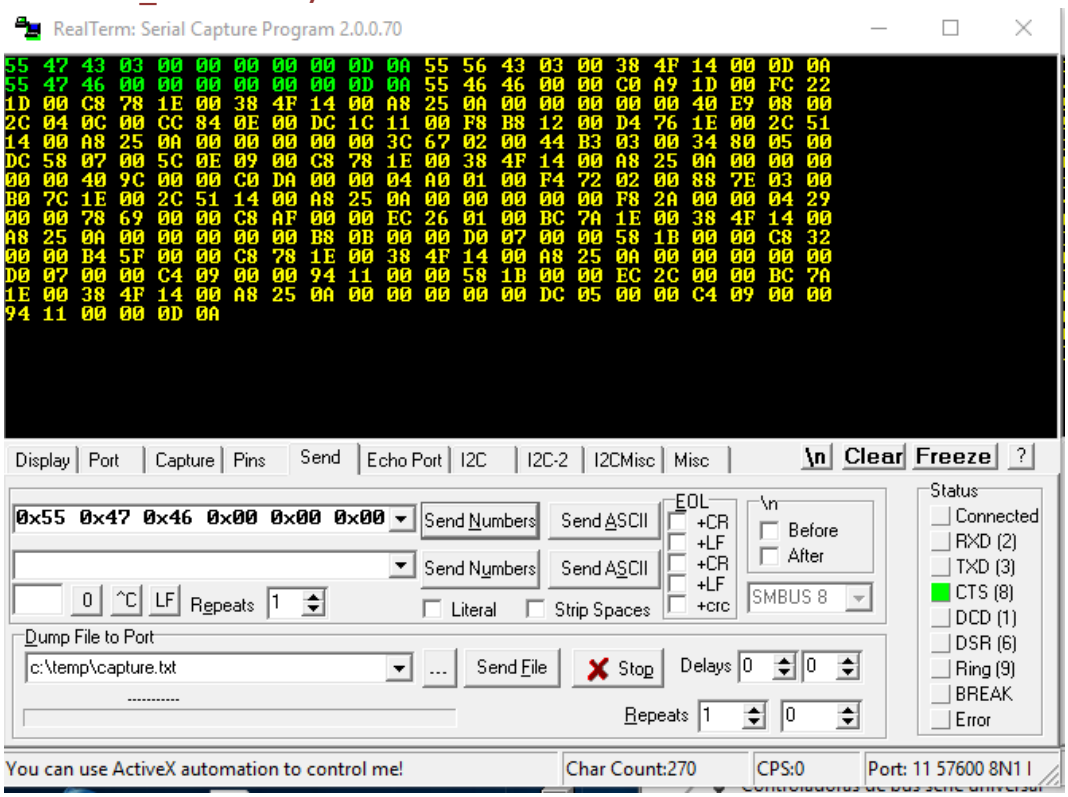

Figure 13: GET_CURRENT y GET_FRAME Examples

In the previous image, the first command corresponds to the request of the value of the photodiode (0,3), with a value 0x144F38 (in the image 38 4F 14 00).

The second command asks for the entire frame and we can see that that the value of the photodiode (0,3) is identical in both commands.

4.7 RESET Command

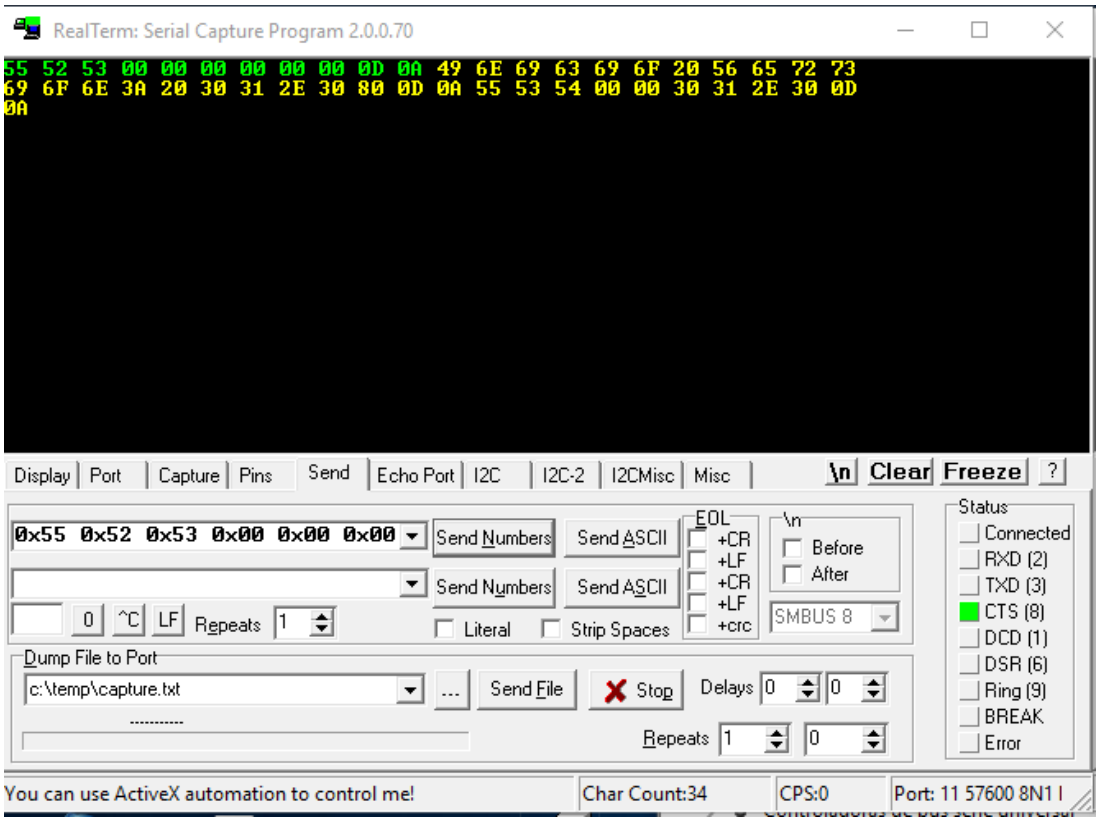

Figure 14: RESET command and START Command in hex format