BASIC PRINCIPLES OF THE COMMUNICATION PROTOCOL

All communication is initiated by the Raspberry as all the devices are connected on the same RS485 bus. The Arduinos are capable of signaling a request to talk by using the signaling lines. If the Raspberry permits a request, it will send an ACK to the corresponding slave, and it will expect an immediate response. The Raspberry will consider that the communication timed out if it does not get a response 20ms after it sent the ACK.

The following Arduinos are implemented in this communication protocol

Imm slave: the Arduino interfacing between the robot and the injection molding machine

Servo slave: the Arduino operating the servo motors driving the robot.

Zmod slave: the Arduino handling the gripper at the end of the Z axis.

The slave addresses are as follow:

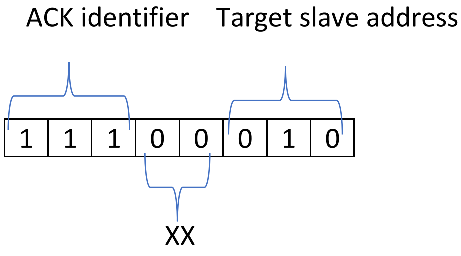
Imm slave	: 0b001
Servo slave	: 0b010
Zmod slave	: 0b011

Message length varies from 1 up to 4 bytes.

MASTER PERMITING SLAVE TO RESPOND

When the master permits a slave to respond, he has to send an ACK message to the specific slave. The ACK message is a single byte. Bits 7 to 5 are always 0b111 and bits 2:0 correspond to the targeted slave, according to their address listed above. Bits 4 and 3 are not used and can have any value.

For example, if the Servo slave wants to transmit an update for the position of the X axis, it will signal the master that can respond with 0b11100010, and the slave can then transmit the update.



MASTER TO SLAVE COMMUNICATION

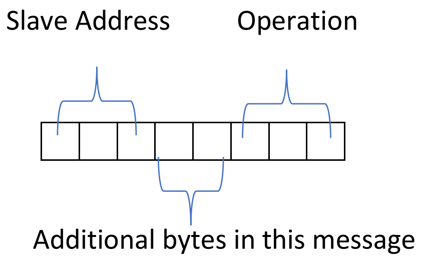
When the Raspberry (Master) sends a message to an Arduino (Slave), other than an ACK message, the first byte contains information about the slave it wants to address, the length of this message and the operation that it wishes to do.

Master to Slave:

bits 7:5 slave address

bits 4:3 the number of additional bytes this message has. The additional bytes can be 0 up to 3, depending on the targeted slave and the operation, as explained below.

bits 2:0 requested operation, specific to the targeted slave as explained below.



AVAILABLE OPERATIONS OF ARDUINO SLAVES

Operations for imm slave

0 0 0 : IMM status report

0 0 1 : Set relays

0 1 0 : Placeholder

0 1 1 : Placeholder

1 0 0 : Placeholder

1 0 1 : Repeat last response

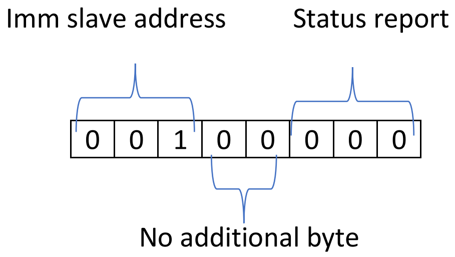
1 1 0 : Placeholder

1 1 1 : Placeholder

Imm status report operation (0b000):

The master requests the status of the injection molding machine (inputs and outputs). There are no additional bytes for this message.

Master transmits 0b00100000



Set relays operation (0b001):

Set the output relays to the injection molding machine. There is one additional byte required for this message.

Master transmits 0b00101001 + 1 additional byte

Byte 2 structure:

bit 7 Not Usable (Relay channel 8)

bit 6 Permit Mold Open (Relay channel 7)

bit 5 Permit Ejector Forward (Relay channel 6)

bit 4 Permit Ejector Retract (Relay channel 5)

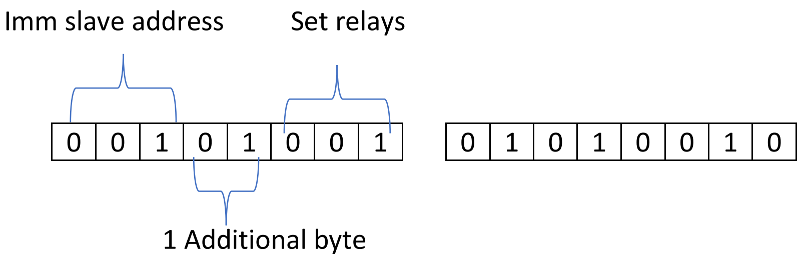
bit 3 Robot Non-Operational (Relay channel 4)

bit 2 Emergency Stop (Relay channel 3)

bit 1 Mold Area Free (Relay channel 2)

bit 0 Permit Mold Close (Relay channel 1)

Example message 0b00101001 0b01010010



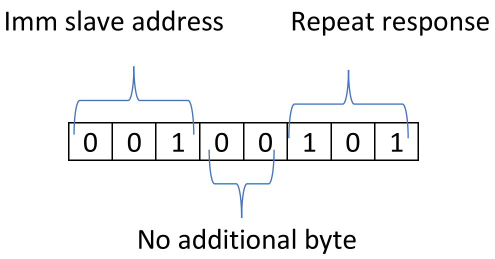
This will set the Permit Mold Open, Permit Ejector Forward and Mold Area Free relays, and reset all the others.

Repeat last response operation(0b101):

The master requests the imm slave to repeat its last message (used if the message was lost

or corrupted). There are no additional bytes for this message.

Master transmits 0b00100101



Operations for servo slave

0 0 0 : Servo status report

0 0 1 : Program

0 1 0 : Next move

0 1 1 : Move axis

1 0 0 : Set servo mode

1 0 1 : Repeat last response

1 1 0 : Zero axis

1 1 1 : Stop

Servo status operation(0b000):

The master requests information about the Servo Arduino. There is one additional byte required for this message that specifies the type of information requested.

Master transmits 0b01001000 + 1 additional Byte

Byte 2 structure:

bits 7:3 : Placeholders

bit 2 1 0 report selection

0 0 0 : report servo status

0 0 1 : report x position

0 1 0 : report y position

0 1 1 : report z position

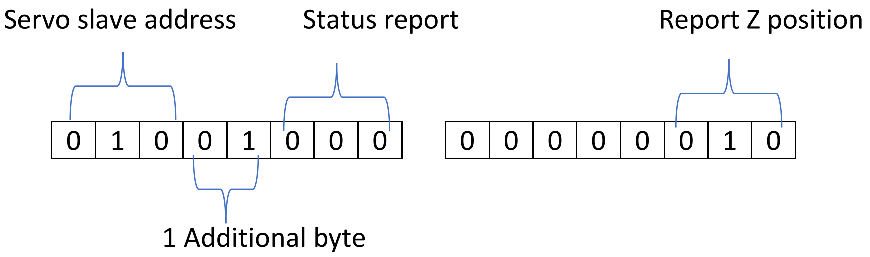
1 0 0 : report saved parameters in EEPROM

1 0 1 : report servo mode

1 1 0 : report current index

1 1 1 : RESERVED

Example message 0b01001000 0b00000010



This will request the current position of the Z axis.

Program operation(0b001):

Used to program move sequences, change saved parameters of the robot or changing the current index during automatic moves. There are additional bytes depending on the programming operation selected.

Programming a move sequence on the Servo slave:

The master first transmits how many move commands it will send (up to 60 moves supported). After specifying the amount of moves that will be transmitted, the master can start transmitting the moves. First, it has to transmit the index of the command, then the move information and last it has to send information regarding to when the next move can start (move delay). Master's transmitions alternate between index, move and next move delay info until he transmits all the commands. Programmed moves are not stored during the Arduino shutdown, and have to be programmed before entering an automatic mode of operation.

Updating a move sequence on the Servo slave:

If the master has already transmitted move commands, he can update individual commands by sending the command index, followed by the move and next move delay info.

Master transmits 0b010yy001 + yy parameter Bytes

program operation identifiers (used in byte 2 to specify what type of program operation the master is accessing):

Byte 2

bit 7 6 5

0 0 0 : Declare number of move commands

0 0 1 : Select programming command index

0 1 0 : Move info for selected programming command index

0 1 1 : Delay info for selected programming command index

1 0 0 : Set parameter

1 0 1 : Set current command index

1 1 0 : Placeholder

1 1 1 : Placeholder

Bytes for program operation identifier 000 (Declare number of move commands), yy = 10.

First command sent when programming a set of moves for the robot. Currently the Servo slave supports up to 60 move commands. 2 additional bytes.

Byte 2 structure:

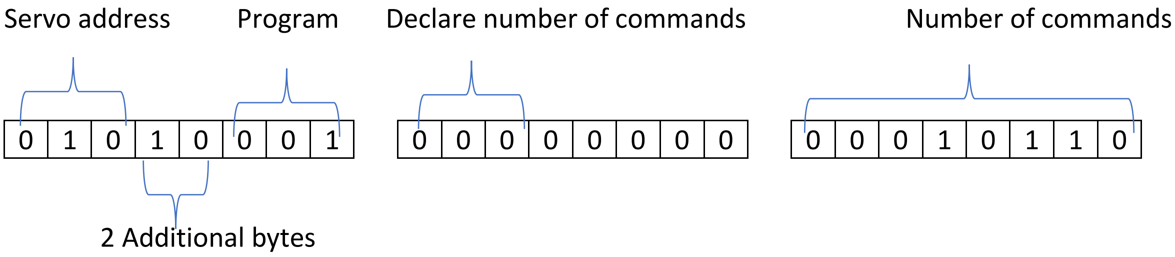
bits 7:5 program operation identifier (000)

bits 4:0 Placeholders

Byte 3 structure:

bits 7:0 number of move commands to be transmitted

Example message 0b01010001 + 0b00000000 + 0b00010110



This will inform the slave that the master wants to program 22 move commands.

Bytes for program operation identifier 001 (Select programming command index), yy = 10.

Selects the index of the command that will be programmed to the robot. Move commands index starts at 0. 2 additional bytes.

Byte 2 structure:

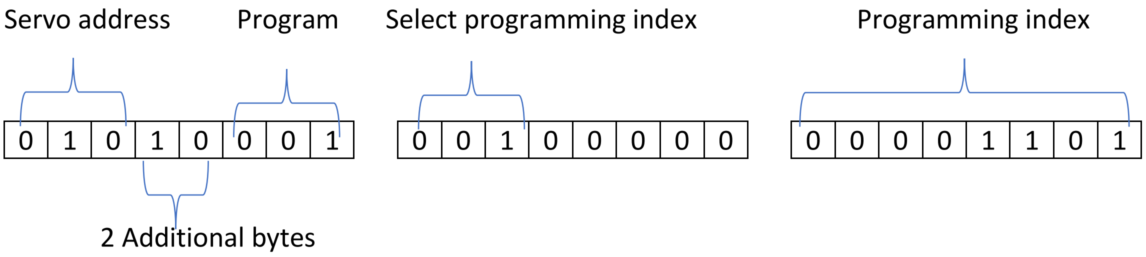
bits 7:5 program operation identifier (001)

bits 4:0 Placeholders

Byte 3 structure:

bits 7:0 selected command index

Example message 0b01001001 0b00100000 0b00001101



This will select 13 as the index of the move to be programmed.

Bytes for program operation identifier 010 (Move info), yy = 11.

This message contains information about the move that will be programmed at the selected programming index. It informs the slave about the axis that will be moved, the end position and the speed of the move. Bytes 2, 3 and 4 follow a similar structure to the Move axis command explained later. 3 additional bytes.

Byte 2 structure:

bits 7:5 program operation identifier (010)

bits 4:0 according to Move axis operation bits (see below)

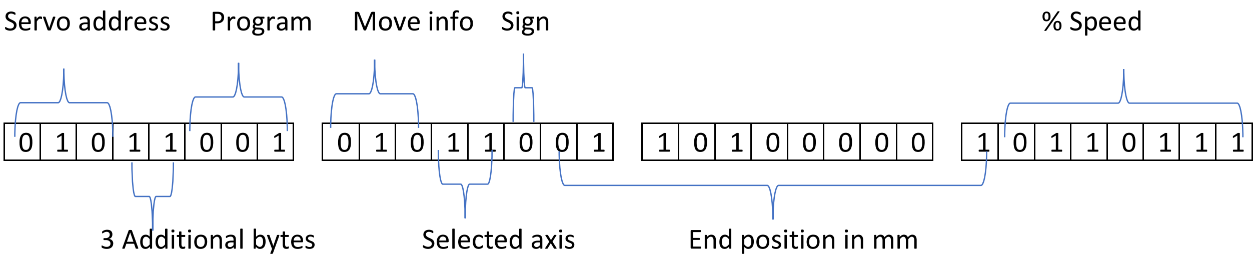
Byte 3 structure:

according to Move axis operation bits (see below)

Byte 4 structure:

according to Move axis operation bits (see below)

Example message 0b01011001 0b01011001 0b10100000 0b10110111



This will program a move for Z axis, with end position at +833 and 55% speed at the selected programming index.

Bytes for program operation identifier 011 (Move delay), yy = 10.

Selects when the robot will perform the next move. If Move delay is set to 1201, the robot will stop after completing this move and wait for a Next move message. Otherwise, the robot will start the next move when it reaches the position specified by this command in mm. 2 additional bytes.

Byte 2 structure:

bits 7:5 program operation identifier (011)

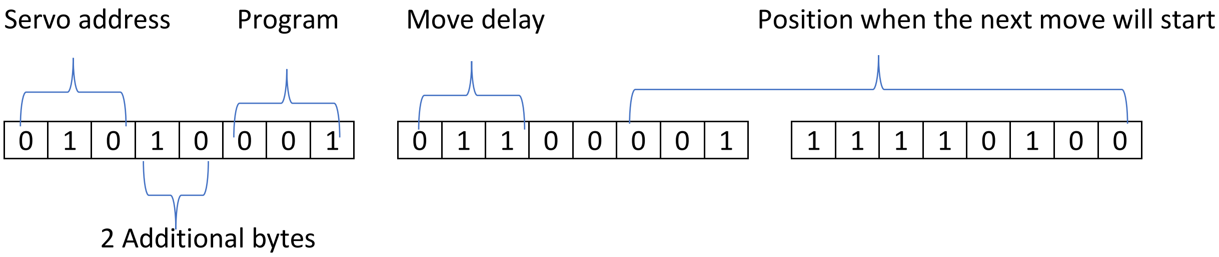
bits 4:3 placeholder

bits 2:0 delay top 3 bits

Byte 3 structure:

bits 7:0 delay lower 8 bits.

Example message 0b01010001 0b01100001 0b11110100



Informs the slave that the next move has to automatically start when the Z axis reaches 500mm.

Bytes for program operation identifier 100 (Set parameter), yy = 11.

Set value to a robot parameter stored in EEPROM (parameters available listed below). 3 additional bytes.

Byte 2 structure:

bits 7:5 program operation identifier (100)

bits 4:0 Parameter index (0 - 31)

Byte 3 structure:

bits 7:0 Parameter value upper 8 bits

Byte 4 structure:

bits 7:0 Parameter value lower 8 bits

List of parameter indexes

0 - (0b00000)	Software version integer part -- 1 byte
1 - (0b00001)	Software version fractional part -- 1 byte
2 - (0b00010)	Servo slave ID -- 1 byte
3 – (0b00011)	Zeroing speed (% of max speed) -- 1 byte
4 – (0b00100)	X axis gear ratio -- 1 byte
5 – (0b00101)	Y axis gear ratio -- 1 byte
6 – (0b00110)	Z axis gear ratio -- 1 byte
7 – (0b00111)	Axes without movement complete feedback pin (flags, bit 2: z axis, bit 1: y axis, bit 0: x axis) -- 1 byte
8 – (0b01000)	X axis pulses per revolution -- 2 bytes
9 – (0b01001)	Y axis pulses per revolution -- 2 bytes

10 – (0b01010) Z axis pulses per revolution -- 2 bytes

11 – (0b01011) Max servo RPM -- 2 bytes

12 – (0b01100) X axis mm per pulley rotation -- 2 bytes

13 – (0b01101) Y axis mm per pulley rotation -- 2 bytes

14 – (0b01110) Z axis mm per pulley rotation -- 2 bytes

15 – (0b01111) X axis length in mm -- 2 bytes

16 – (0b10000) Y axis length in mm -- 2 bytes

17 – (0b10001) Z axis length in mm -- 2 bytes

18 – (0b10010) X axis ms delay for movement complete signal if pin is missing -- 2 bytes

19 – (0b10011) Y axis ms delay for movement complete signal if pin is missing -- 2 bytes

20 – (0b10100) Z axis ms delay for movement complete signal if pin is missing -- 2 bytes

21 – (0b10101) Placeholder

22 – (0b10110) Placeholder

23 – (0b10111) Placeholder

24 – (0b11000) Placeholder

25 – (0b11001) Placeholder

26 – (0b11010) Placeholder

27 – (0b11011) Placeholder

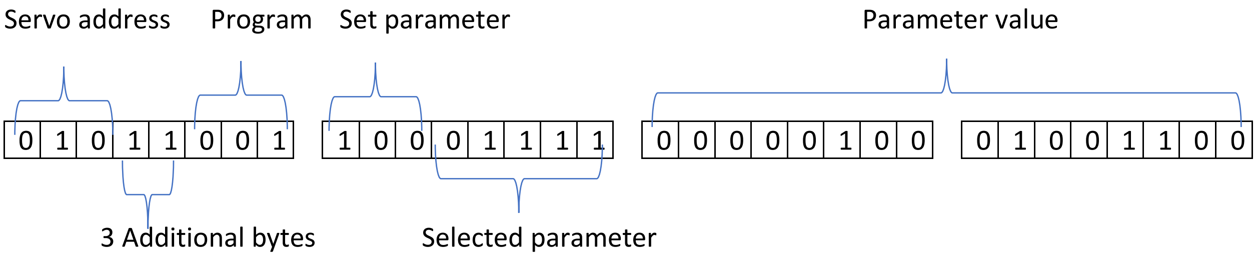
28 – (0b11100) Placeholder

29 – (0b11101) Placeholder

30 – (0b11110) Placeholder

31 – (0b11111) Placeholder

Example message 0b01011001 0b10001111 0b00000100 0b01001100



This will set the length of the Z axis to 1100mm.

Bytes for program operation identifier 101 (Set current command index), yy = 10.

This command will set the current command index of the automatic mode. 2 additional bytes.

Byte 2 structure:

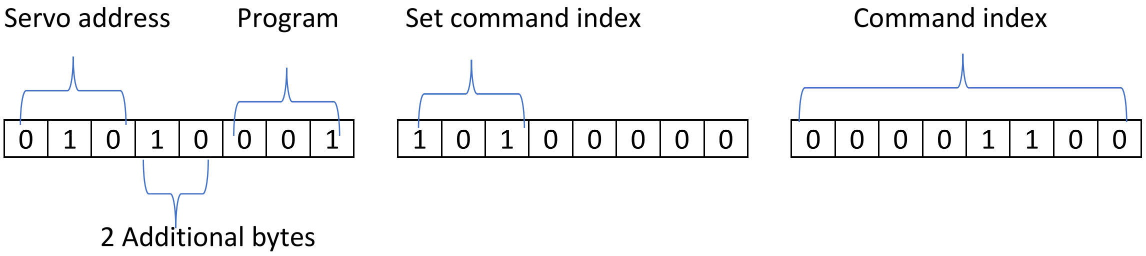
bits 7:5 program operation identifier (101)

bits 4:0 Placeholders

Byte 3 structure:

bits 7:0 selected current command index

Example message 0b01010001 0b10100000 0b00001100

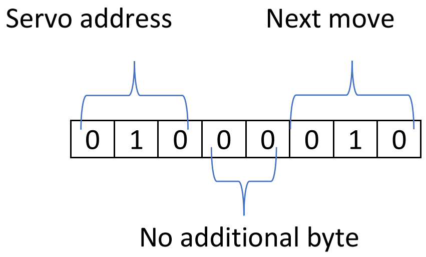


Sets the command index for the automatic mode to 12.

Next move operation(0b010):

If the robot is in automatic mode, increments the current command index causing the robot to perform the next set of moves. There are no additional bytes for this message.

Master transmits 0b01000010



Move axis operation(0b011):

Moves an axis to the specified position (absolute position) with a specific percent speed (in regards to the max RPM of the servos set for the robot). The servo Arduino checks if the requested move is within bounds with regards to the axis length specified for the robot. Maximum supported position is 2047mm, and the speed can be set up to 127%. 3 additional bytes.

Master transmits 0b01011011 + 3 additional Bytes

Byte 2 structure:

bits 7:5 Placeholders

bit 4:3 Axis (00 invalid , 01 x axis, 10 y axis , 11 z axis)

bit 2 Sign bit (0 positive mm , 1 negative mm)

bits 1:0 Position bits 10:9

Byte 3 structure:

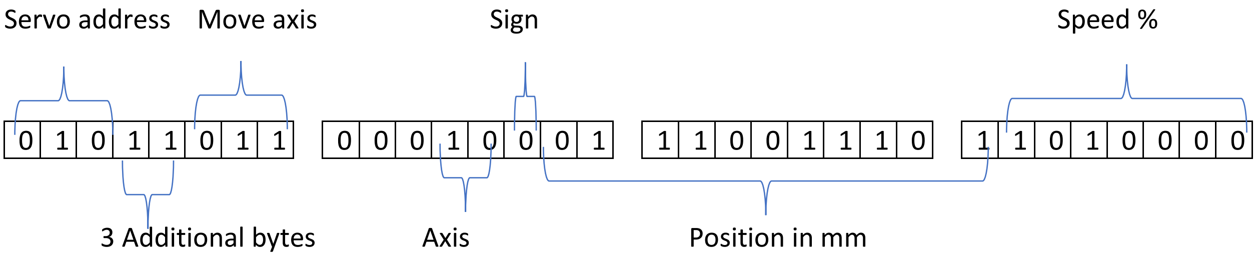
bits 7:0 Position bits 8:1

Byte 4 structure:

bit 7 Position bit 0

bits 6:0 Speed

Example message 0b01011011 0b00010001 0b11001110 0b11010000



This will move the Y axis at +925mm with 80% speed.

Set servo mode operation(0b100):

Sets the mode of the servo slave. 1 additional byte required.

Master transmits 0b01001100 + 1 additional Byte

List of modes

0-00 Manual operation mode. All axes have to be zeroed before any move command is sent. Servo cannot accept Next move operations.

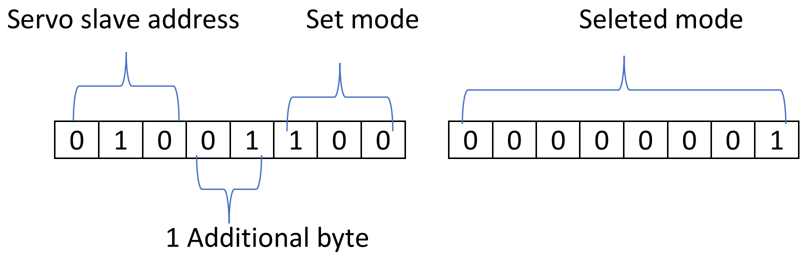
1-01 Automatic operation mode, servo can only accept Next move, stop and set mode operations. Cannot set mode to automatic if the axes haven't zeroed or if the axes are not at the starting position according to the programmed move sequence.

2-10 Service operation mode, the servo can accept move commands without zeroing the axes first. While in service mode the servo will attempt any move instructed without checking for safe conditions like running out of bounds of an axis (unlike manual and automatic mode) , use with caution.

Byte 2 structure:

bits 7:0 : Servo mode

Example message 0b01001100 0b00000001

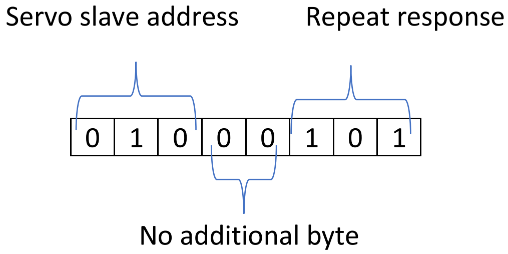


This will set the mode to Automatic.

Repeat last response operation(0b101):

The master requests the servo slave to repeat its last message (used if the message was lost or corrupted). There are no additional bytes for this message.

Master transmits 0b01000101



Zero axis operation(0b110):

Zeroes the axes selected by the flags of the second byte. 1 additional byte.

Master transmits 0b01001110 + 1 additional Byte

Byte 2 structure:

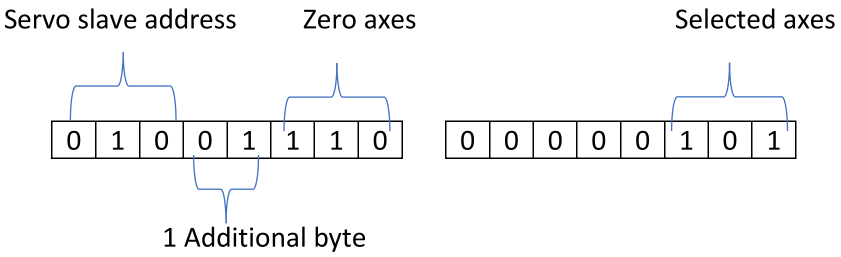
bit 7:3 : Placeholders

bit 2 : Zero z axis

bit 1 : Zero y axis

bit 0 : Zero x axis

Example message 0b01001110 0b00000101

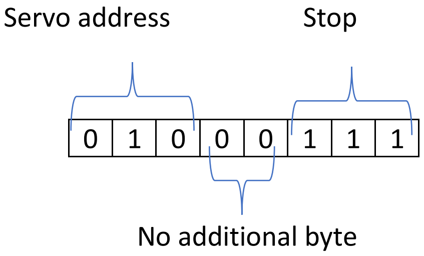


This will command axes Z and X to zero.

Stop operation(0b111):

Stops any current move of the robot axes. No additional bytes.

Master transmits 0b01000111



Operations for Zmod slave

0 0 0 : Report status

0 0 1 : Set Zmod outputs

0 1 0 : Placeholder

0 1 1 : Placeholder

1 0 0 : Placeholder

1 0 1 : Repeat last response

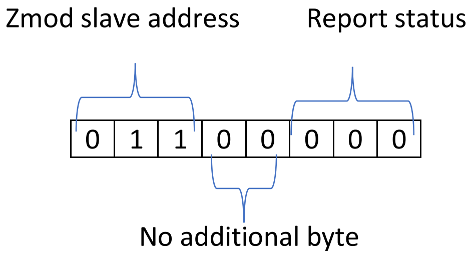
1 1 0 : Placeholder

1 1 1 : Placeholder

Zmod status report operation (0b000):

The master requests the status of the Zmod module. There are no additional bytes for this message.

Master transmits 0b01100000



Set Zmod outputs (0b001):

Set the output relays of the Zmod module. There is one additional byte required for this message.

Master transmits 0b01101001 + 1 additional byte

Byte 2 structure:

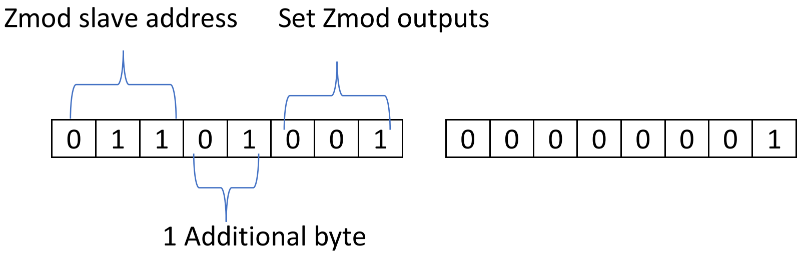
bits 7:3 : Placeholders

bit 2 : Output 3 (Not connected)

bit 1 : Output 2 (Not connected)

bit 0 : Output 1 (Gripper – Suction cup relay)

Example message 0b01101001 0b00000001



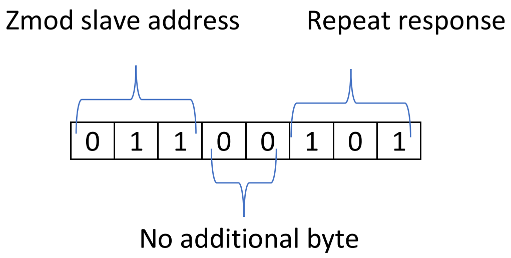
This will set the gripper – suction relay, and reset all the others.

Repeat last response operation(0b101):

The master requests the Zmod slave to repeat its last message (used if the message was lost

or corrupted). There are no additional bytes for this message.

Master transmits 0b01100101



SLAVE TO MASTER COMMUNICATION

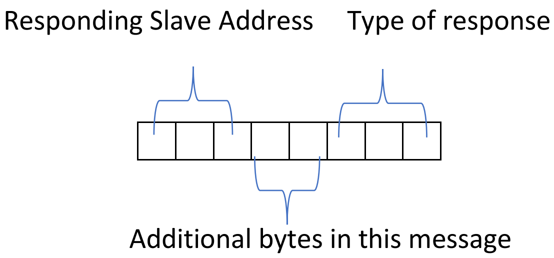
When the slave gets an ACK after signaling the master that it wants to transmit , it can respond. The first byte contains information about the responding slave, the length of the message and the type of response that it's transmitting.

Slave to Master:

bits 7:5 responding slave address

bits 4:3 the number of additional bytes this message has. The additional bytes can be 0 up to 3, depending on the targeted slave and the operation, as explained below.

bits 2:0 type of response, specific to the targeted slave as explained below.



Responses for imm slave

0 0 0 : IMM status response

IMM status response(0b000):

Informs the master about the status of the imm slave. 2 additional bytes.

Slave transmits 0b00110000 + 2 additional Bytes

Byte 2 structure:

bit 7 Relay channel 8 status (Not usable)

bit 6 Relay channel 7 status (Permit Mold Open)

bit 5 Relay channel 6 status (Permit Ejector Forward)

bit 4 Relay channel 5 status (Permit Ejector Retract)

bit 3 Relay channel 4 status (Robot Non-Operational)

bit 2 Relay channel 3 status (Emergency Stop)

bit 1 Relay channel 2 status (Mold Area Free)

bit 0 Relay channel 1 status (Permit Mold Close)

Byte 3 structure:

bit 7 Signal 7 status (Mold Fully Closed) (Active low)

bit 6 Signal 6 status (Fully Automatic) (Active low)

bit 5 Signal 5 status (Ejector Fully Forward) (Active low)

bit 4 Signal 4 status (Ejector Fully Retracted) (Active low)

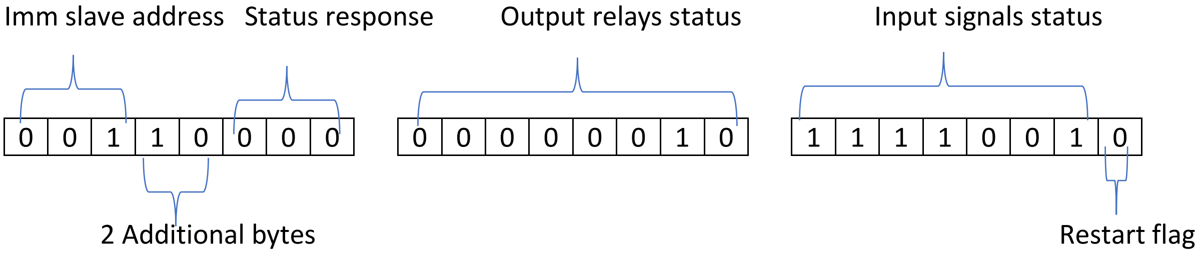
bit 3 Signal 3 status (Movable Gates Closed) (Active low)

bit 2 Signal 2 status (Mold Fully Open) (Active low)

bit 1 Signal 1 status (Emergency Stop) (Active low)

bit 0 Restart flag (sent once when the Arduino powers up)

Example message 0b00110000 0b00000000 0b00000000



This message informs the master that the Mold area free relay is set and that the injection molding machine is signaling that the movable gates are closed and the mold is fully open.

Response operation for servo slave

0 0 0 : report servo status

0 0 1 : report x position

0 1 0 : report y position

0 1 1 : report z position

1 0 0 : report saved parameters in EEPROM

1 0 1 : report servo mode

1 1 0 : report current index

1 1 1 : report auto move completion

Servo status response operation(0b000):

Informs the master about the status of the servo slave. 3 additional bytes.

Slave transmits 0b01011000 + 3 additional Bytes

Byte 2 structure (servo errors):

bit 7 Automatic move sequence has not been set up yet, robot will not be able to enter automatic mode

bit 6 Servo board has restarted (sent once when the Arduino powers up) bit 5 Axes have not been zeroed yet, cannot perform any move other than zeroing

bit 4 Requested move aborted because axis is getting pulses and moving

bit 3 Servo not at the correct coordinates for entering automatic mode

bit 2 Invalid servo mode for requested command

bit 1 Requested move out of axis bounds

bit 0 Active alarm on servo (DC servo supply has to be turned off and back on manually)

Byte 3 structure (servo status high byte):

bit 7 Next command bit flag (Not meant to be read, used by the slave to start automatic moves)

bit 6 Z axis currently zeroing bit flag

bit 5 Y axis currently zeroing bit flag

bit 4 X axis currently zeroing bit flag

bit 3 Z axis direction bit flag (0 CW, 1CCW)

bit 2 Y axis direction bit flag (0 CW, 1CCW)

bit 1 X axis direction bit flag (0 CW, 1CCW)

bit 0 Z axis inductive bit flag (1 when the inductive sensor triggers)

Byte 4 structure (servo status low byte):

bit 7 Y axis inductive bit flag (1 when the inductive sensor triggers)

bit 6 X axis inductive bit flag (1 when the inductive sensor triggers)

bit 5 Z axis move ended bit flag (See note below)

bit 4 Y axis move ended bit flag (See note below)

bit 3 X axis move ended bit flag (See note below)

bit 2 Z axis move started bit flag (See note below)

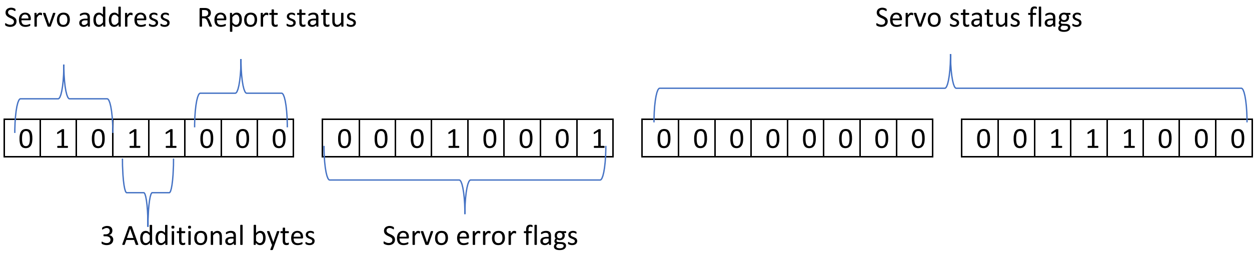
bit 1 Y axis move started bit flag (See note below)

bit 0 X axis move started bit flag (See note below)

Note: Axis move ended bits do not correspond to the state of the servo because the movement complete pins of the servo motors trigger constantly during acceleration and deceleration. The state of the servo can be determined by checking both axis move started and ended bits for each axis as follows:

State	Move started bit	Move ended bit
Axis is idle	0	1
Axis is getting pulses	1	1
Pulses stopped, axis is slowing down	0	0
Movement ended(idle)	0	1

Example message 0b01011000 0b00010001 0b00000000 0b00111000



0	1	0	1	1	0	0	0
0	0	0	1	0	0	0	1
0	0	0	0	0	0	0	0
0	0	1	1	1	0	0	0

This message informs the master about the current errors (alarm on servos and move aborted) and the status of the servo module (all axes are idle).

X position response operation(0b001):

Informs the master about the X axis position (mm). 2 additional bytes.

Slave transmits 0b01010001 + 2 additional Bytes

Byte 2 structure:

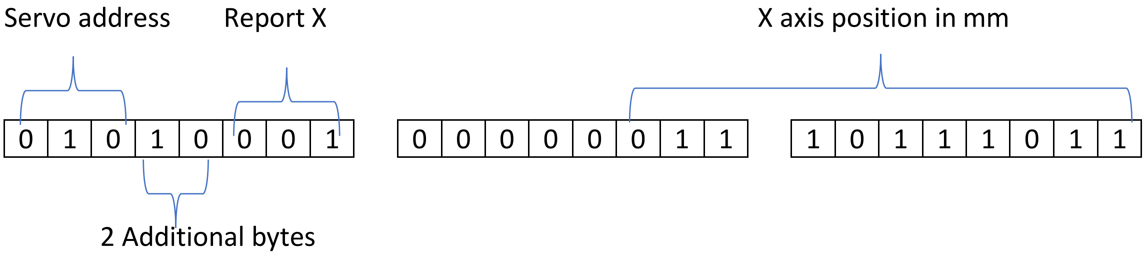
bits 7:3 Placeholders

bits 2:0 X Position bits 10:8

Byte 3 structure:

bits 7:0 X Position bits 7:0

Example message 0b01010001 0b00000011 0b10111011



0 1 0 1 0 0 0 1   0 0 0 0 0 0 1 1   1 0 1 1 1 0 1 1

This message informs the master that the X axis is at 955mm.

Y position response operation(0b010):

Informs the master about the Y axis position (mm). 2 additional bytes.

Slave transmits 0b01010010 + 2 additional Bytes

Byte 2 structure:

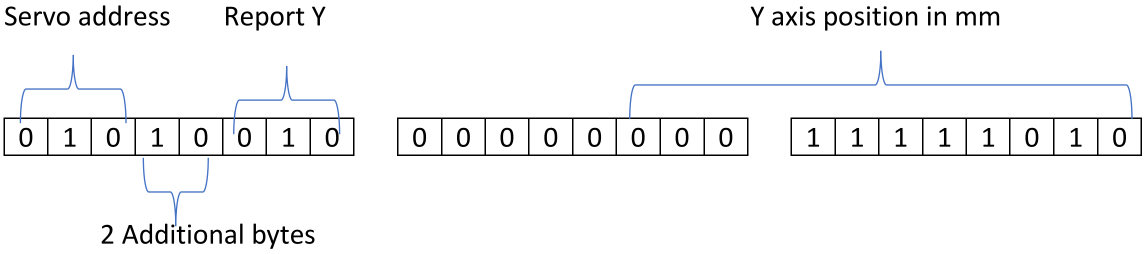
bits 7:3 Placeholders

bits 2:0 Y Position bits 10:8

Byte 3 structure:

bits 7:0 Y Position bits 7:0

Example message 0b01010010 0b00000000 0b11111010



0	1	0	1	0	0	1	0
0	0	0	0	0	0	0	0
1	1	1	1	1	0	1	0

This message informs the master that the Y axis is at 250mm.

Z position response operation(0b011):

Informs the master about the Z axis position (mm). 2 additional bytes.

Slave transmits 0b01010011 + 2 additional Bytes

Byte 2 structure:

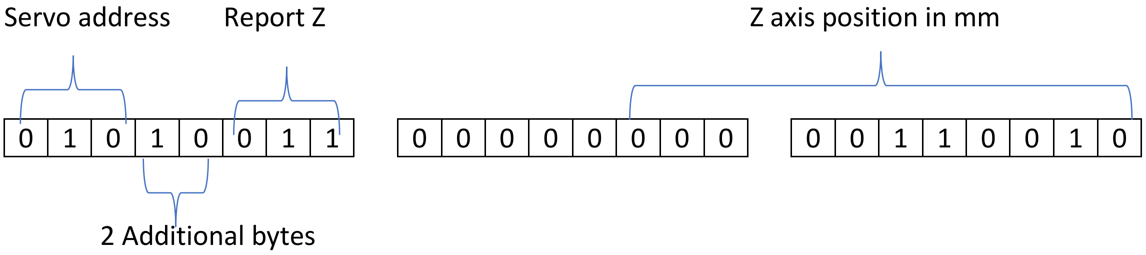
bits 7:3 Placeholders

bits 2:0 Z Position bits 10:8

Byte 3 structure:

bits 7:0 Z Position bits 7:0

Example message 0b01010011 0b00000000 0b00110010



0	1	0	1	0	0	1	1
0	0	0	0	0	0	0	0
0	0	1	1	0	0	1	0

This message informs the master that the Z axis is at 50mm.

Report saved parameters in EEPROM operation(0b100):

Transmits all the saved robot parameters (as listed previously) to the master. This will transmit one message per parameter. 3 additional bytes per message.

Slave transmits 0b01011100 + 3 additional Bytes

Byte 2 structure:

bits 7:5 Placeholders

bits 4:0 Parameter index

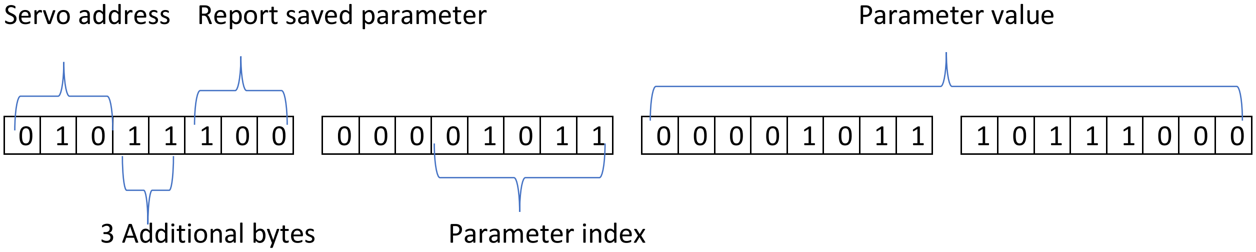
Byte 3 structure:

bits 7:0 Parameter value high byte

Byte 4 structure:

bits 7:0 Parameter value low byte

Example message 0b01011100 0b00001011 0b00001011 0b10111000



This message informs the master that the listed RPM of the servo motors is 3000RPM.

Report servo mode operation(0b101):

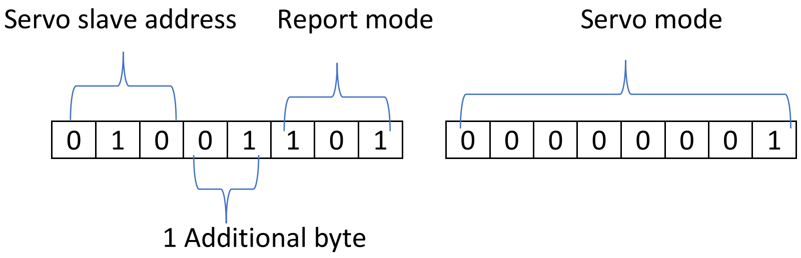
Transmits the mode that the servo module is currently in. 1 additional byte.

Slave transmits 0b01001101 + 1 additional Byte

Byte 2 structure:

bits 7:0 servo mode

Example message 0b01001101 0b00000001



0	1	0	0	1	1	0	1	0	0	0	0	0	0	0	1

This informs the master that the servo module is in automatic mode.

Report current index operation(0b110):

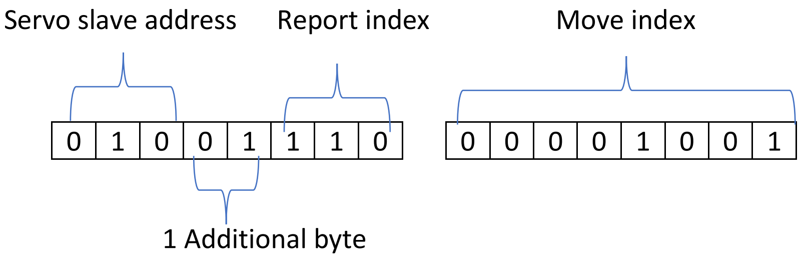
Transmits the current move index during automatic mode operation. 1 additional byte.

Slave transmits 0b01001110 + 1 additional Byte

Byte 2 structure:

bits 7:0 current command index

Example message 0b01001110 0b00001001



0	1	0	0	1	1	1	0	0	0	0	0	1	0	0	1

This informs the master that the current index of automatic moves is 9.

Report auto move operation(0b111):

Informs the master about axes starting or completing an automatic sequence move, as well as the index of this move. 2 additional bytes.

Slave transmits 0b01010111 + 2 additional Bytes

Byte 2 structure:

bits 7:3 Placeholders

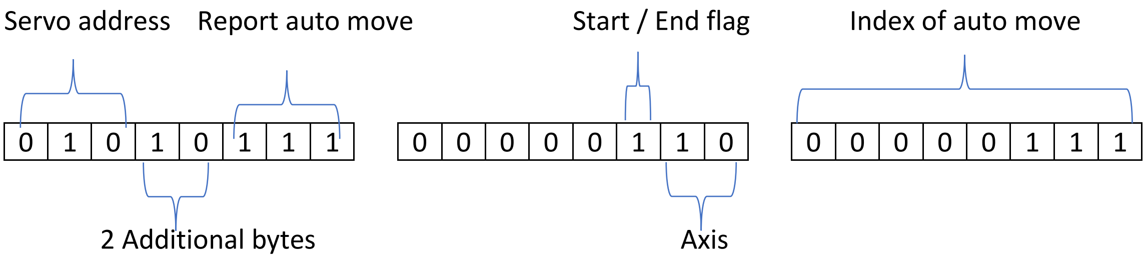
bit 2: 0 if the axis started an automatic move, 1 if the axis completed it

bits 1:0 Axis

Byte 3 structure:

bits 7:0 Starting index of the auto move

Example message 0b01010111 0b00000110 0b00000111



0	1	0	1	0	1	1	1	0	0	0	0	0	1	1	0	0	0	0	0	0	1	1	1

This message informs the master that the Y axis completed the automatic move with index 7.

Response operation for z mod slave

0 0 0 : Report status

Zmod status response(0b000):

Informs the master about the status of the Zmod slave. 1 additional byte.

Slave transmits 0b01101000 + 1 additional Byte

Byte 2 structure:

bit 7 Sample timeout (Not meant to be read - flag used for sampling input pins)

bit 6 Input 3 state (Active low)

bit 5 Input 2 state (Active low)

bit 4 Input 1 state (Gripper feedback) (Active low)

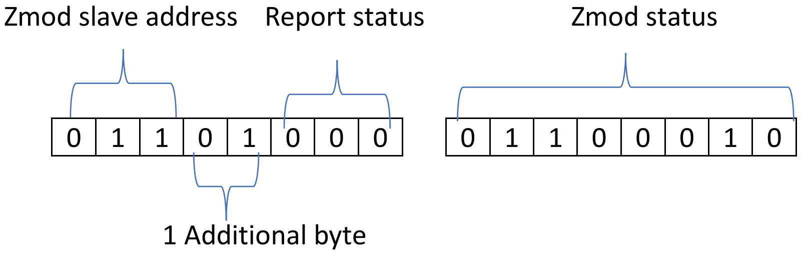
bit 3 Output 3 state

bit 2 Output 2 state

bit 1 Output 1 state (Gripper relay)

bit 0 Restart flag (sent once when the Arduino powers up)

Example message 0b01101000 0b01100010



This informs the master that the gripper relay is set and that there is feedback from the gripper.