

## iMicro V20 I2C Command Guide for Motion Control

### Introduction

The iMicro V20 has one motion control module mounted on board. The module is connected to the I2C bus as a slave (address = 0x50). It supports both read and write. Note that the module may stretch the clock if necessary during read transaction. This document provides details of the I2C commands for motion control.

## Motion Control I2C Command Guide

### I2C Write Command Sequence

S	I2C Address	A	Register	A	Length	A	Data0	A	Data1	A	....	Data n-1	A	P
---	-------------	---	----------	---	--------	---	-------	---	-------	---	------	----------	---	---

S	Start Condition
I2C Address	I2C address of the slave, with R/W bit = 0
A	Acknowledge from slave
Register	Register address to written. The register address for motion control is <b>0x20</b>
Length	Length of the data to be written to slave. <b>Note:</b> The maximum length is 127. The MSB of the length byte is 0 for writing data. The MSB of the length byte is 1 for reading data.
Data0 – Data n-1	The data bytes (assume the length is n)
P	Stop condition

### I2C Read Command Sequence

S	I2C Address	A	Register	A	Length	A	RS	Data0	Am	Data1	Am	....	Data n-1	NAm	P
---	-------------	---	----------	---	--------	---	----	-------	----	-------	----	------	----------	-----	---

S	Start Condition
I2C Address	I2C address of the slave, with R/W bit = 0
A	Acknowledge from slave
Register	Register address to written. The register address for motion control is <b>0x20</b>
Length	Length of the data to be read from slave. <b>Note:</b> The maximum length is 127. The MSB of the length byte is 0 for writing data. The MSB of the length byte is 1 for reading data.
RSAm	Re-start condition
Data0 – Data n-1	The data bytes (assume the length is n)
Am	Acknowledge from Master
NAm	No Acknowledge from Master
P	Stop condition

## Motion Control Command Data Format

The data packet to write to the motion controller and read from motion controller should follow the following format.

CMD_TYPE	Parameter Bytes	0x00
----------	-----------------	------

CMD_TYPE	1 byte to indicate the command to be used
Data Bytes	The parameters for this command. The length varies for different commands
0x00	Indicates the end of the command packet

When writing to the motion controller, the data packet is formed and the length is calculated. The length and data packet is then sent in the I2C write command sequence to send to motion controller.

When reading from the motion controller, the expected length of data is sent in the I2C read sequence. Once the data packet is received, the data can be interpreted using the command format.

The command type and parameters are shown in the table below.

## Motion Control Command to Change Motor State

To change the motor state, the corresponding command packet is to be written to the motion controller using the Write Command Sequence.

### Change Velocity (mm/sec) of Motor 0 and Motor 1

CMD_TYPE	0x00					
I2C Write Packet	0x00	Velocity 0	Velocity 1	0x00		
Data Length (byte)	10					
Note:	Velocity data is 4 byte each, IEEE-754 floating point format					

### Change Position (mm) of Motor 0 and Motor 1

CMD_TYPE	0x01					
I2C Write Packet	0x01	Position 0	Position1	0x00		
Data Length (byte)	10					
Note:	Position data is 4 byte each, IEEE-754 floating point format					

### Change PWM duty cycle of driver of Motor 0 and Motor 1

CMD_TYPE	0x08					
I2C Write Packet	0x08	PWM 0	PWM 1	0x00		
Data Length (byte)	10					
Note:	PWM data is 4 byte each, IEEE-754 floating point format					

### Change Acceleration Limit (mm/sec/sec) of Motor 0 and Motor 1

CMD_TYPE	0x02					
I2C Write Packet	0x02	Acceleration 0	Acceleration 1	0x00		
Data Length (byte)	10					
Note:	Acceleration data is 4 byte each, IEEE-754 floating point format					

### Change Velocity Limit (mm/sec) for position mode of Motor 0 and Motor 1

CMD_TYPE	0x03					
I2C Write Packet	0x03	Velocity 0	Velocity 1	0x00		
Data Length (byte)	10					
Note:	Velocity data is 4 byte each, IEEE-754 floating point format					

### Change PID parameters for Motor0

CMD_TYPE	0x04					
I2C Write Packet	0x04	Kp	Ki	Kd	0x00	
Data Length (byte)	14					
Note:	PID data is 4 byte each, IEEE-754 floating point format					

### Change PID parameters for Motor1

CMD_TYPE	0x06					
I2C Write Packet	0x06	Kp	Ki	Kd	0x00	
Data Length (byte)	14					
Note:	PID data is 4 byte each, IEEE-754 floating point format					

**Change Wheel parameters for Motor0**

CMD_TYPE	0x05					
I2C Write Packet	0x05	Gear Ratio	Wheel Diameter	CPR	0x00	
Data Length (byte)	14					
Note:	All wheel data is 4 byte each, IEEE-754 floating point format					

**Change Wheel parameters for Motor1**

CMD_TYPE	0x07					
I2C Write Packet	0x07	Gear Ratio	Wheel Diameter	CPR	0x00	
Data Length (byte)	14					
Note:	All wheel data is 4 byte each, IEEE-754 floating point format					

**Change Velocity (mm/sec) of Motor 0**

CMD_TYPE	0x10					
I2C Write Packet	0x10	Velocity 0	0x00			
Data Length (byte)	6					
Note:	Velocity data is 4 byte each, IEEE-754 floating point format					

**Change Velocity (mm/sec) of Motor 1**

CMD_TYPE	0x20					
I2C Write Packet	0x20	Velocity 1	0x00			
Data Length (byte)	6					
Note:	Velocity data is 4 byte each, IEEE-754 floating point format					

**Change Position (mm) of Motor 0**

CMD_TYPE	0x11					
I2C Write Packet	0x11	Position 0	0x00			
Data Length (byte)	6					
Note:	Position data is 4 byte each, IEEE-754 floating point format					

**Change Position (mm) of Motor 1**

CMD_TYPE	0x21					
I2C Write Packet	0x21	Position 1	0x00			
Data Length (byte)	6					
Note:	Position data is 4 byte each, IEEE-754 floating point format					

**Change PWM duty cycle of driver of Motor 0**

CMD_TYPE	0x14					
I2C Write Packet	0x14	PWM 0	0x00			
Data Length (byte)	6					
Note:	PWM data is 4 byte each, IEEE-754 floating point format					

**Change PWM duty cycle of driver of Motor 1**

CMD_TYPE	0x24					
I2C Write Packet	0x24	PWM 1	0x00			
Data Length (byte)	6					
Note:	PWM data is 4 byte each, IEEE-754 floating point format					

**Change Acceleration Limit (mm/sec/sec) of Motor 0**

CMD_TYPE	0x12				
I2C Write Packet	0x12	Acceleration 0	0x00		
Data Length (byte)	6				
Note:	Acceleration data is 4 byte each, IEEE-754 floating point format				

**Change Acceleration Limit (mm/sec/sec) of Motor 1**

CMD_TYPE	0x22				
I2C Write Packet	0x22	Acceleration 1	0x00		
Data Length (byte)	6				
Note:	Acceleration data is 4 byte each, IEEE-754 floating point format				

**Change Velocity Limit (mm/sec) for position mode of Motor 0**

CMD_TYPE	0x13				
I2C Write Packet	0x13	Max Velocity 0	0x00		
Data Length (byte)	6				
Note:	Velocity data is 4 byte each, IEEE-754 floating point format				

**Change Velocity Limit (mm/sec) for position mode of Motor 1**

CMD_TYPE	0x13				
I2C Write Packet	0x13	Max Velocity 1	0x00		
Data Length (byte)	6				
Note:	Velocity data is 4 byte each, IEEE-754 floating point format				

**Emergency Stop for Motor 0 and Motor 1**

CMD_TYPE	0x70				
I2C Write Packet	0x70	0x00			
Data Length (byte)	2				
Note:	No parameter byte needed				

**Emergency Stop for Motor 0**

CMD_TYPE	0x71				
I2C Write Packet	0x71	0x00			
Data Length (byte)	2				
Note:	No parameter byte needed				

**Emergency Stop for Motor 1**

CMD_TYPE	0x72				
I2C Write Packet	0x72	0x00			
Data Length (byte)	2				
Note:	No parameter byte needed				

**Perform Reset for Motor 0 and Motor 1**

CMD_TYPE	0x73				
I2C Write Packet	0x73	0x00			
Data Length (byte)	2				
Note:	No parameter byte needed				

**Perform Reset for Motor 0**

CMD_TYPE	0x74				
I2C Write Packet	0x74	0x00			
Data Length (byte)	2				
Note:	No parameter byte needed				

**Perform Reset for Motor 1**

CMD_TYPE	0x75				
I2C Write Packet	0x75	0x00			
Data Length (byte)	2				
Note:	No parameter byte needed				

**Turn OFF power of Motor 0 and Motor 1**

CMD_TYPE	0x77				
I2C Write Packet	0x77	0x00			
Data Length (byte)	2				
Note:	No parameter byte needed				

**Turn ON power of Motor 0 and Motor 1**

CMD_TYPE	0x78				
I2C Write Packet	0x78	0x00			
Data Length (byte)	2				
Note:	No parameter byte needed				

**Turn OFF power for Motor 0**

CMD_TYPE	0x79				
I2C Write Packet	0x79	0x00			
Data Length (byte)	2				
Note:	No parameter byte needed				

**Turn ON power for Motor 0**

CMD_TYPE	0x7A				
I2C Write Packet	0x7A	0x00			
Data Length (byte)	2				
Note:	No parameter byte needed				

**Turn OFF power for Motor 1**

CMD_TYPE	0x7B				
I2C Write Packet	0x7B	0x00			
Data Length (byte)	2				
Note:	No parameter byte needed				

**Turn ON power for Motor 1**

CMD_TYPE	0x7C				
I2C Write Packet	0x7C	0x00			
Data Length (byte)	2				
Note:	No parameter byte needed				

## Motion Control Command to Read Motor State

To read the motor state, the corresponding command packet is to be written to the motion controller using the Write Command Sequence first, then the Read Command Sequence is performed to read the data back.

### Read Velocity (mm/sec) of Motor 0 and Motor 1

CMD_TYPE	0x80				
I2C Write Packet	0x80	0x00			
Data Length (byte)	2				
Read Data Packet	0x80	Velocity 0	Velocity1	0x00	
Data Length (byte)	10				
Note:	Velocity data is 4 byte each, IEEE-754 floating point format				

### Read Position (mm) of Motor 0 and Motor 1

CMD_TYPE	0x81				
I2C Write Packet	0x81	0x00			
Data Length (byte)	2				
Read Data Packet	0x81	Position 0	Position 1	0x00	
Data Length (byte)	10				
Note:	Position data is 4 byte each, IEEE-754 floating point format				

### Read PWM Duty Cycle of Motor 0 and Motor 1

CMD_TYPE	0x88				
I2C Write Packet	0x88	0x00			
Data Length (byte)	2				
Read Data Packet	0x88	PWM 0	PWM 1	0x00	
Data Length (byte)	10				
Note:	Velocity data is 4 byte each, IEEE-754 floating point format				

### Read Acceleration Limit of Motor 0 and Motor 1

CMD_TYPE	0x82				
I2C Write Packet	0x82	0x00			
Data Length (byte)	2				
Read Data Packet	0x82	Acceleration 0	Acceleration 1	0x00	
Data Length (byte)	10				
Note:	Acceleration data is 4 byte each, IEEE-754 floating point format				

### Read Velocity Limit (mm/sec) for position mode of Motor 0 and Motor 1

CMD_TYPE	0x83				
I2C Write Packet	0x83	0x00			
Data Length (byte)	2				
Read Data Packet	0x83	Velocity 0	Velocity 1	0x00	
Data Length (byte)	10				
Note:	Velocity data is 4 byte each, IEEE-754 floating point format				

**Read PID parameters of Motor 0**

CMD_TYPE	0x84				
I2C Write Packet	0x84	0x00			
Data Length (byte)	2				
Read Data Packet	0x84	Kp	Ki	Kd	0x00
Data Length (byte)	14				
Note:	PID data is 4 byte each, IEEE-754 floating point format				

**Read PID parameters of Motor 1**

CMD_TYPE	0x86				
I2C Write Packet	0x86	0x00			
Data Length (byte)	2				
Read Data Packet	0x86	Kp	Ki	Kd	0x00
Data Length (byte)	14				
Note:	PID data is 4 byte each, IEEE-754 floating point format				

**Read Wheel parameters of Motor 0**

CMD_TYPE	0x85				
I2C Write Packet	0x85	0x00			
Data Length (byte)	2				
Read Data Packet	0x85	Gear Ratio	Wheel Diameter	CPR	0x00
Data Length (byte)	14				
Note:	Wheel data is 4 byte each, IEEE-754 floating point format				

**Read Wheel parameters of Motor 1**

CMD_TYPE	0x87				
I2C Write Packet	0x87	0x00			
Data Length (byte)	2				
Read Data Packet	0x85	Gear Ratio	Wheel Diameter	CPR	0x00
Data Length (byte)	14				
Note:	Wheel data is 4 byte each, IEEE-754 floating point format				

**Read Velocity (mm/sec) of Motor0**

CMD_TYPE	0x90				
I2C Write Packet	0x90	0x00			
Data Length (byte)	2				
Read Data Packet	0x90	Velocity 0	0x00		
Data Length (byte)	6				
Note:	Velocity data is 4 byte each, IEEE-754 floating point format				

**Read Velocity (mm/sec) of Motor1**

CMD_TYPE	0xA0				
I2C Write Packet	0xA0	0x00			
Data Length (byte)	2				
Read Data Packet	0xA0	Velocity 0	0x00		
Data Length (byte)	6				
Note:	Velocity data is 4 byte each, IEEE-754 floating point format				



**Read Position (mm) of Motor 0**

CMD_TYPE	0x91				
I2C Write Packet	0x91	0x00			
Data Length (byte)	2				
Read Data Packet	0x91	Position 0	0x00		
Data Length (byte)	6				
Note:	Position data is 4 byte each, IEEE-754 floating point format				

**Read Position (mm) of Motor 1**

CMD_TYPE	0xA1				
I2C Write Packet	0xA1	0x00			
Data Length (byte)	2				
Read Data Packet	0xA1	Position 1	0x00		
Data Length (byte)	6				
Note:	Position data is 4 byte each, IEEE-754 floating point format				

**Read Acceleration Limit (mm/sec/sec) of Motor 0**

CMD_TYPE	0x92				
I2C Write Packet	0x92	0x00			
Data Length (byte)	2				
Read Data Packet	0x92	Acceleration 0	0x00		
Data Length (byte)	6				
Note:	Acceleration data is 4 byte each, IEEE-754 floating point format				

**Read Acceleration Limit (mm/sec/sec) of Motor 1**

CMD_TYPE	0xA2				
I2C Write Packet	0xA2	0x00			
Data Length (byte)	2				
Read Data Packet	0xA2	Acceleration 1	0x00		
Data Length (byte)	6				
Note:	Acceleration data is 4 byte each, IEEE-754 floating point format				

**Read Velocity Limit (mm/sec) for position mode of Motor 0**

CMD_TYPE	0x93				
I2C Write Packet	0x93	0x00			
Data Length (byte)	2				
Read Data Packet	0x93	Velocity 0	0x00		
Data Length (byte)	6				
Note:	Velocity data is 4 byte each, IEEE-754 floating point format				

**Read Velocity Limit (mm/sec) for position mode of Motor 1**

CMD_TYPE	0xA3				
I2C Write Packet	0xA3	0x00			
Data Length (byte)	2				
Read Data Packet	0xA3	Velocity 1	0x00		
Data Length (byte)	6				
Note:	Velocity data is 4 byte each, IEEE-754 floating point format				

### Read PWM Duty Cycle of Motor 0

CMD_TYPE	0x94				
I2C Write Packet	0x94	0x00			
Data Length (byte)	2				
Read Data Packet	0x94	PWM 0	0x00		
Data Length (byte)	6				
Note:	PWM data is 4 byte each, IEEE-754 floating point format				

### Read PWM Duty Cycle of Motor 1

CMD_TYPE	0xA4				
I2C Write Packet	0xA4	0x00			
Data Length (byte)	2				
Read Data Packet	0xA4	PWM 1	0x00		
Data Length (byte)	6				
Note:	PWM data is 4 byte each, IEEE-754 floating point format				

### Example to change motor speed

iMoto Address: 0x70

To set: Motor 1 speed = 112.3 mm/sec, Motor 2 speed = -234.5 mm/sec

112.3 in hex is 0x42E0999A, -234.5 in hex is 0xC36A8000

Command Packet to sent to the iMoto

0x00	0x9A	0x99	0xE0	0x42	0x00	0x80	0x6A	0xC3	0x00
------	------	------	------	------	------	------	------	------	------

- Send start condition
- Send I2C address 0x70
- Send register address 0x20
- Send length 0x0A
- Send the command packet above
- Send Stop condition

### Example to Read motor Position

iMoto Address: 0x70

Command Packet to sent to the iMoto

0x81	0x00
------	------

Read back data from iMoto is as follows (assume Motor0 = 112.3mm, Motor 1 is -234.5mm)

0x81	0x9A	0x99	0xE0	0x42	0x00	0x80	0x6A	0xC3	0x00
------	------	------	------	------	------	------	------	------	------

## I2C Command Sequence

- Send start condition
- Send I2C address 0x70
- Send register address 0x20
- Send length 0x02
- Send the command packet above
- Send Stop condition
  
- Send start condition
- Send I2C address 0x71
- Send register address 0x20
- Send length 0x0A
- Send Restart Condition
- Read 10 byte of data (Acknowledge for the first 9 bytes, NO-ACK for the last byte)
- Send Stop condition