ME218b Smart Product Design

Offensive Coordinator for ME 218b Project 2011

Purpose:

The primary purpose of the Offensive Coordinator is to act as a gateway to the field infrastructure to allow your robot to request that a ball be released, request that the hoop diameter be changed, query the state of a ball dispenser and query the state of the game.

Interface Connection

Connector:

The connector of the Offensive Coordinator is a 6-pin keyed Molex connector.

Pinout:

Pin	Name/Function
1	+5V (@ 100mA) / Power to the Offensive Coordinator(V_{dd})
2	SDI / Serial Data Into the Offensive Coordinator
3	SDO / Serial Data Out of the Offensive Coordinator
4	SCK / Serial Clock
5	SS / active low select line for the Offensive Coordinator
6	GND / Ground reference for the Offensive Coordinator

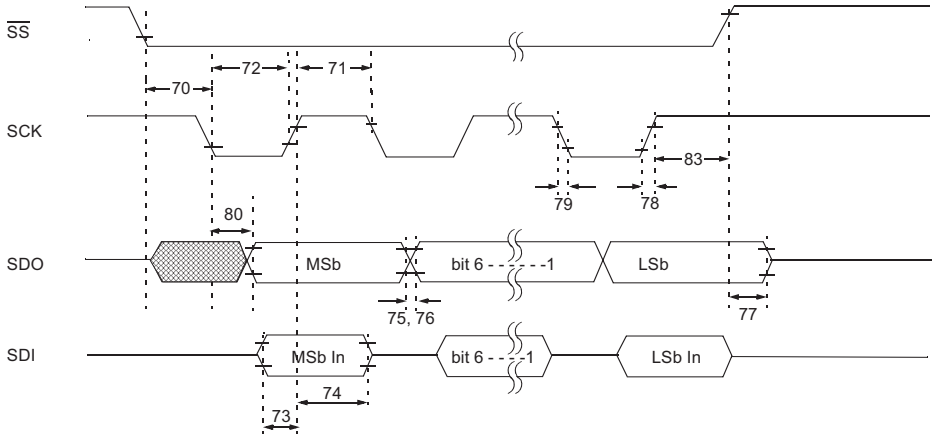
Electrical Specifications

Parameter	Min.	Max	Units
V_{iH}	V_{dd}*0.65		V
V_{oH}	V_{dd}-0.4		V
V_{iL}		V_{dd}*0.35	V
V_{oL}		0.4	V
$I_{iH,}$ I_{iL}		±1	µA
I_{oH}	-20		µA
I_{OL}	20		µA
All Specifications at V_{dd} = 5V			

Byte Transfer Specification

The Offensive Coordinator uses a synchronous serial signaling method to transfer data into and out of the Offensive Coordinator. The signaling method is compatible with SPI communications, with the Offensive Coordinator operating as a slave device on an SPI network. The $\overline{SS}$ line must be lowered to begin a byte transfer and raised at the completion of the byte transfer.

The relationships between the four lines involved in the transfer of a byte are shown in the figure & table below:



Param No.	Symbol	Characteristic		Min	Typ	Max	Units	Conditions
70*	TssL2scH, TssL2scL	$\overline{SS}\downarrow$ to SCK$\downarrow$ or SCK$\uparrow$ input		Tcy[a]	—	—	ns	
71*	TscH	SCK input high time (Slave mode)		Tcy + 20	—	—	ns	
72*	TscL	SCK input low time (Slave mode)		Tcy + 20	—	—	ns	
73*	TdiV2scH, TdiV2scL	Setup time of SDI data input to SCK edge		100	—	—	ns	
74*	TscH2diL, TscL2diL	Hold time of SDI data input to SCK edge		100	—	—	ns	
75*	TdoR	SDO data output rise time	3.0-5.5V	—	10	25	ns	
			2.0-5.5V	—	25	50	ns	
76*	TdoF	SDO data output fall time		—	10	25	ns	
77*	TssH2doZ	$\overline{SS}\uparrow$ to SDO output high-impedance		10	—	50	ns	
78*	TscR	SCK output rise time (Master mode)	3.0-5.5V	—	10	25	ns	
			2.0-5.5V	—	25	50	ns	
79*	TscF	SCK output fall time (Master mode)		—	10	25	ns	
80*	TscH2doV, TscL2doV	SDO data output valid after SCK edge	3.0-5.5V	—	—	50	ns	
			2.0-5.5V	—	—	145	ns	
83*	TscH2ssH, TscL2ssH	$\overline{SS}\uparrow$ after SCK edge		1.5Tcy + 40	—	—	ns	

* These parameters are characterized but not tested. [a]Tcy = 33μS

Byte Protocol Specification

Common Byte Format:

Exchanges between the Offensive Coordinator(OC) and your robot take place with two successive bytes being exchanged. The first byte from the robot to the OC is the actual command. Except in the case of the reception of an invalid command byte (covered below) the value returned from the OC during this transfer has no meaning. The robot must wait a minimum of 1mS before sending the second byte of the exchange. The value sent to the OC as the second byte of the sequence has no meaning while the value returned by the second byte transfer is the result from the command byte. In the tables below, only the meaningful commands and replies are shown, not the two meaningless values that are part of the exchanges.

Robot to Offensive Coordinator Bytes:

The meaningful values for the command bytes from the robot to the Offensive Coordinator are shown in the following table:

Command	Meaning
0b0100 00xx	Request Ball be dispensed from dispenser xx
0x4F	Request Hoop to 3pt size
0x48	Request Hoop to 2pt size
0x74	Query status of last request
0x78	Query game status
0b0111 00xx	Query balls available from dispenser xx

Offensive Coordinator to Robot Bytes:

The values and meanings of the response bytes returned by the Offensive Coordinator are shown in the following table:

Response Byte	Description of meaning
0b1011 xx00	Query of dispenser xx pending
0b1100 xxyy	Dispenser xx has yy balls available
0xD1	Last request is pending (response to request status query)
0xDF	Last request succeeded (response to request status query)
0xD0	Last request failed (response to request status query)
0xE0	Game is paused (response to game status query)
0xE1	Game is running (response to game status query)
0xE2	Game is over, Red won (response to game status query)
0xE4	Game is over, Green won (response to game status query)
0xFE	Invalid Command Byte Received

Any response byte that is received that is not listed in this table should be treated as if it had the value 0xFF.

Request that a Ball be Dispensed from a Dispenser:

To request that a ball be dispensed from a dispenser, send a byte of 0b0111 00*xx* to the Offensive Coordinator, replacing *xx* with the number of the dispenser from which balls will be dispensed. The Offensive Coordinator will immediately reply with 0xD1, indicating that the request has been forwarded to the field and the OC is busy waiting for the field's reply. Your 'bot must now query the status of the last request to determine when the ball has actually been dispensed.

Request Hoop to 3 Point Size:

To request that a Hoop be set to the 3-point size, send a byte of 0x4F to the Offensive Coordinator. The Offensive Coordinator will immediately respond with 0xD1, indicating that the request has been forwarded to the field and the OC is busy waiting for the field's reply. Your 'bot must query the status of the last request to determine when the Hoop has actually been re-sized.

Request Hoop to 2 Point Size:

To request that a Hoop be set to the 2-point size, send a byte of 0x48 to the Offensive Coordinator. The Offensive Coordinator will immediately respond with 0xD1, indicating that the request has been forwarded to the field and the OC is busy waiting for the field's reply. Your 'bot must query the status of the last request to determine when the Hoop has actually been re-sized.

Query status of Last Request:

To query the status of the last request made, send a byte of 0x74 to the Offensive Coordinator. The Offensive Coordinator will immediately reply with 0xD1, 0xDF or 0xD0 depending on the current status of the last request.

Query status of the Game:

To query the status of the game, send a byte of 0x78 to the Offensive Coordinator. The Offensive Coordinator will immediately reply with 0xE0 or 0xE1 depending on the current state of the game.

Query Balls Available:

To query the number of balls available from a dispenser, send a byte of 0b0111 00*xx* to the Offensive Coordinator, where *xx* is the number of the dispenser to query. If this is a new request or the status of the last request is not yet known, the Offensive Coordinator will immediately reply with 0b1011 *xx*00, where *xx* is the number of dispenser. You must continue to query the status of the same dispenser until the status has been determined and the Offensive Coordinator replies with 0b1100 *xxyy*, where *xx* is the number of dispenser queried and *yy* is the number of available balls. Once a query of a dispenser has been initiated, no other query or request may be made until the OC returns something other than 0b1011 *xx*00, where *xx* is the number of dispenser queried.

Multiple Request Response:

The response to a second request, while there is a previous request pending, will be 0xFE. The completion of any request is indicated by a query that returns 0xD0 or 0xDF. **It is not possible to simply issue requests without querying the status of the prior request.**

Power on and reset behavior:

Initially, after power on or a reset, the Offensive Coordinator will return 0xFF from any query until such time as the Offensive Coordinator is internally initialized.

Command Timing:

The interval between the issuing of two successive command sequences from robot to Offensive Coordinator must be at least 100ms. This is the interval between successive commands, not the interval between the two bytes of a command sequence.

Invalid Command Bytes:

If the Offensive Coordinator receives a command byte not listed in the table above, it will treat the invalid command byte as if it had not been received and return a value of 0xFE during the next byte exchange with the robot.

Detecting Loss of Synchronization:

Proper operation of the interface requires that the robot and OC remain in synchronization with respect to which are command bytes, dummy bytes and response bytes. While the actual values used as the dummy byte by the robot is technically free, a good choice for that byte would be one that is not a valid command to the OC. In this way, if the OC is expecting a command byte but the robot thinks that it is sending a dummy byte, the OC will detect the dummy byte value as an invalid command and report that on the next transfer. Similarly, the dummy byte returned by the OC to a command has no significance under normal operation but that byte should be monitored by the robot (looking for 0xFE) to detect a loss of synchronization.

Sample Byte Sequence:

'Bot to OC	0x78	0xXX	0x78	0xXX	0x71	0xXX	0x71	0xXX	0x41	0xXX	0x74	0xXX	0x4F	0xXX
OC to 'Bot	0xXX	0xE0	0xXX	0xE1	0xXX	0xB4	0xXX	0xB5	0xXX	0xD1	0xXX	0xDF	0xXX	0xD1

In this sequence, the 'bot queries the state of the game, which will return paused at the beginning of the game. The next game state query returns running as the game state. The 'bot then queries the number of balls available from dispenser 1 and the OC responds that it is busy getting that info. A second query of the number of ball available from dispenser number 1 returns that there is now 1 ball available. The 'bot then requests that a ball be dispensed from dispenser number 1 and the OC replies that the request is pending. Next, a query of the status of the last request indicates that the request succeeded. The final request is for the hoop size to be set to the 3-point diameter to which the immediate reply is that the request is pending.

Physical Specifications

Dimensions:

The Offensive Coordinator dimensions are 2.0" x 3.0" x 1.0".