MIPR

Modular Intelligent Programmable Robot

SDK Specification
Version 1.0.0

HTTPS://L33T.UK/SDK

L33T Components

CONTENTS

ABOUT

This SDK was designed to give engineers more control over MIPR, allowing you to have finite control over the robot and it's sensors using whatever tools you choose.

The SDK will give you a toolbox to control MIPR and acquire data from any of it's sensors, it will even allow you to have direct control over the IO pins attached to the Auxiliary Connector enabling you to create sensor boards and program them without the need to reprogram MIPR.

This is achieved by creating a set of keywords that will be sent to MIPR over Bluetooth. These keywords will give MIPR specific commands. They come in three types;

1. Command Value Pairs
2. Ordinary Commands
3. Command Prompt

Command Value pairs contain two bits of information the first bit being the command and the second bit being the value. These will be separated with a colon, an example of this would be “REF:20”. This command will set the Telemetry Packet refresh rate to 20mS.

The ordinary commands are just that, a keyword that tells the robot to do something. An example would be “SETSEN“ this command will tell MIPR to setup SB002 and initialise the VL53L1X sensor.

Command prompts will be where a command is sent to the robot and then the robot will respond with a question that you will answer.

This document will consist of these commands in table format where I will detail the command, a description and possible return values including errors.

The document will be spilt into sections each section detailing commands available for certain sensor boards finishing with general IO commands.

In order to use the commands laid out in this document MIPR must be set to Op Mode 9. To put MIPR in Op Mode 9 connect to the robot using a terminal program (the serial monitor in Arduino IDE will work) at 57600 baud and type a capitol “O” for Oscar then press enter. The prompt will ask you to enter an Op Mode, type “9” then press enter. MIPR will now restart in Op Mode 9.

When Op Mode 9 is set the robot will do nothing until told. By default it will sit there until a Telemetry Packet is chosen and commands are sent to the robot.

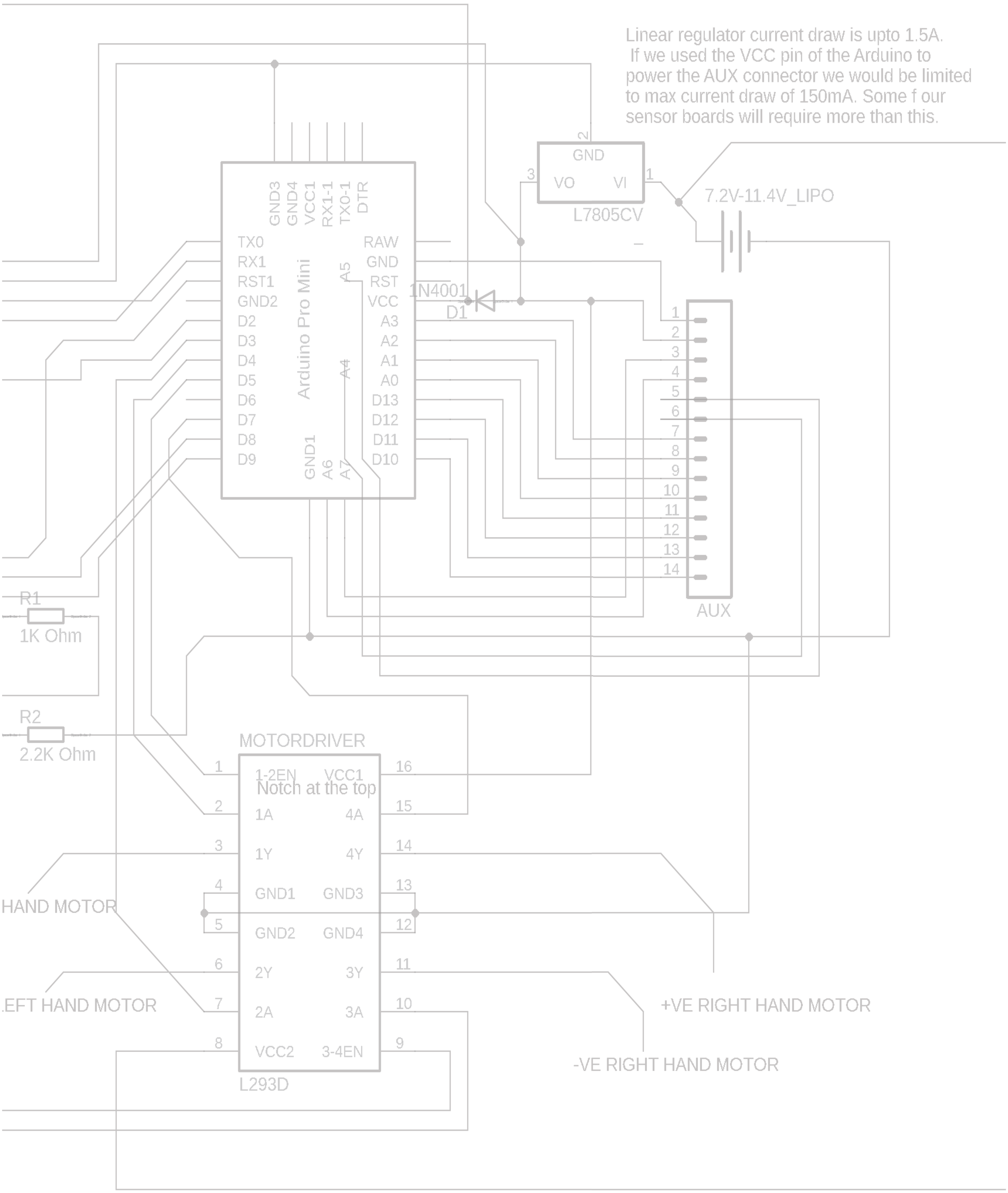
CONNECTING TO MIPR

Once MIPR is in SDK mode Op Mode 9 connect to MIPR over Bluetooth using UART at 57600 baud or 9600 baud if you haven't changed the baud rate on the HC-05. For instructions on how to change the HC-05's baud rate visit https://www.l33t.uk/arduino_projects/mipr/mipr-writing-the-software/

Be aware that MIPR will operate at 57600 baud this means that it can send/receive 57600 bits per second, bare this in mind when setting the packet refresh rate each character will take up 8 bits.

NOTES

D1 stops the bluetooth module and AUX connector from being powered when the Arduino is connected to the computer.



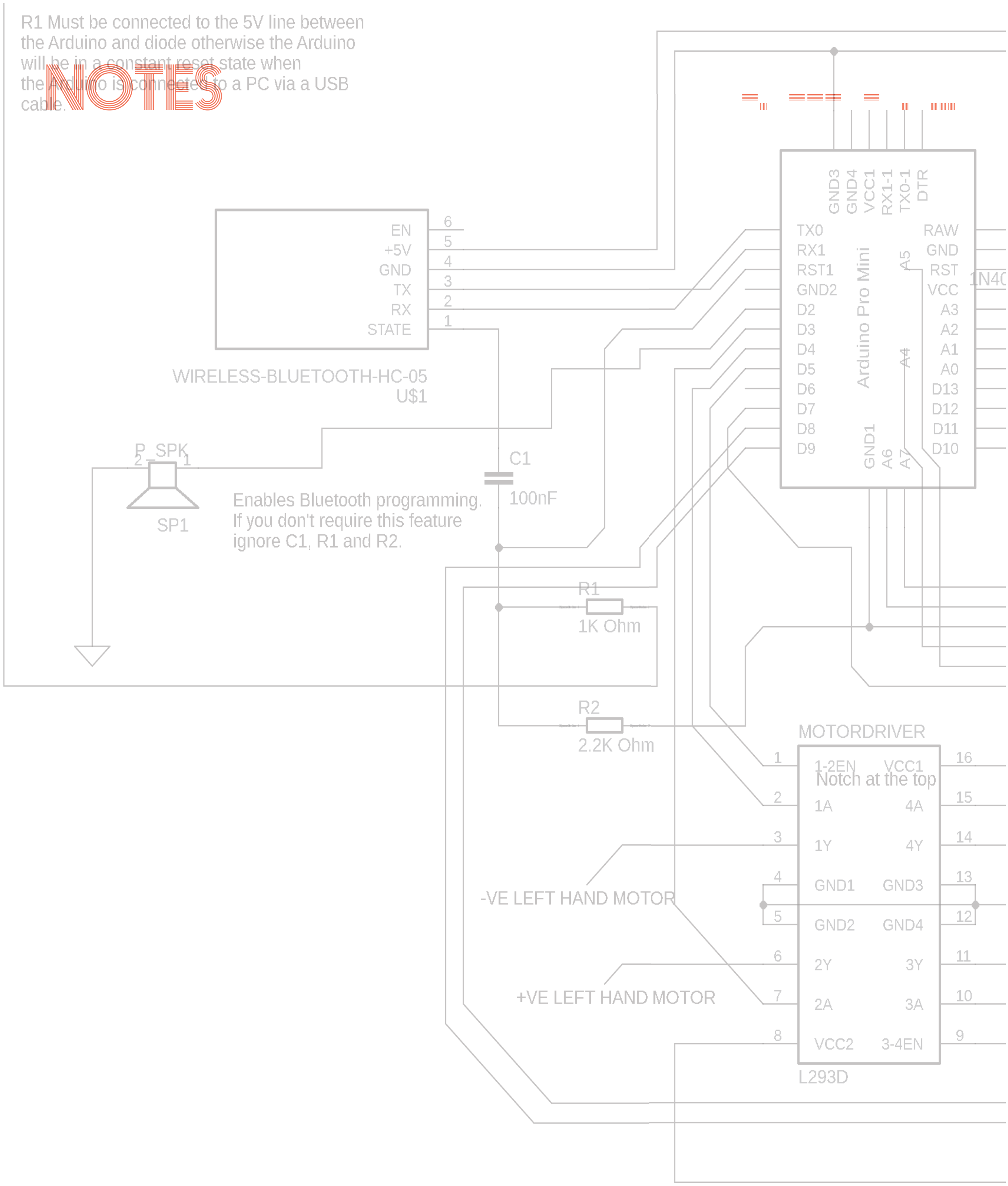
Linear regulator current draw is upto 1.5A. If we used the VCC pin of the Arduino to power the AUX connector we would be limited to max current draw of 150mA. Some f our sensor boards will require more than this.

GENERIC

These commands will be used with the MIPR core robot. No sensor boards are needed for these generic commands. They will allow you to do various core tasks such as making the robot move and setting its speed.

COMMAND	TYPE	DESCRIPTION	BOARD	EXAMPLE	RETURN VALUES
F	Ordinary	Makes the robot move forwards at the set speed	N/A	F	Nil
B	Ordinary	Makes the robot move backwards at the set speed	N/A	B	Nil
S	Ordinary	Stops the robot	N/A	S	Nil
L	Ordinary	Makes the robot move left at the set speed	N/A	L	Nil
R	Ordinary	Makes the robot move right at the set speed	N/A	R	Nil
1 - 9	Ordinary	Set the robots speed in coarse increments. 1 is the slowest and 9 is the fastest	N/A	5	Set the robots speed in increments of 23. 9 is maximum speed
O (for Oscar)	Ordinary Prompt	Sets the Op Mode, once O is entered the program will prompt the user to choose an Op Mode. Once chosen MIPR will restart in that mode	N/A	O then 1	Prompts the users to choose an Op Mode using the following prompt “Enter mode and press enter: “
SON	Ordinary	Turns the speaker on	N/A	SON	Nil
SOFF	Ordinary	Turns the Speaker off	N/A	SOFF	Nil
SPD	Command Value Pair	Sets the robots speed in fine increments	N/A	SPD:200	The value can be in the range of 1 to 255
SLSPD	Command Value Pair	Sets the speed of the left motor	N/A	LSPD:200	The value can be in the range of 1 to 255
SRSPD	Command Value Pair	Sets the speed of the right motor	N/A	RSPD:200	The value can be in the range of 1 to 255
GLSPD	Ordinary	Returns the left motor speed value	N/A	GLSPD	Returns the value that the motor has been set to
GRSPD	Ordinary	Returns the right motor speed value	N/A	GRSPD	Returns the value that the motor has been set to
LSTU	Ordinary	Returns the status of the left motor	N/A	LSTU	RETURNED VALUES WILL BE F, B OR S
RSTU	Ordinary	Returns the status of the right motor	N/A	RSTU	RETURNED VALUES WILL BE F, B OR S
GOM	Ordinary	Return the current Op Mode	N/A	GOM	Returns an integer between 0 and 9

NOTES



R1 Must be connected to the 5V line between the Arduino and diode otherwise the Arduino will be in a constant reset state when the Arduino is connected to a PC via a USB cable.

Enables Bluetooth programming. If you don't require this feature ignore C1, R1 and R2.

Notch at the top

ODOMETRY

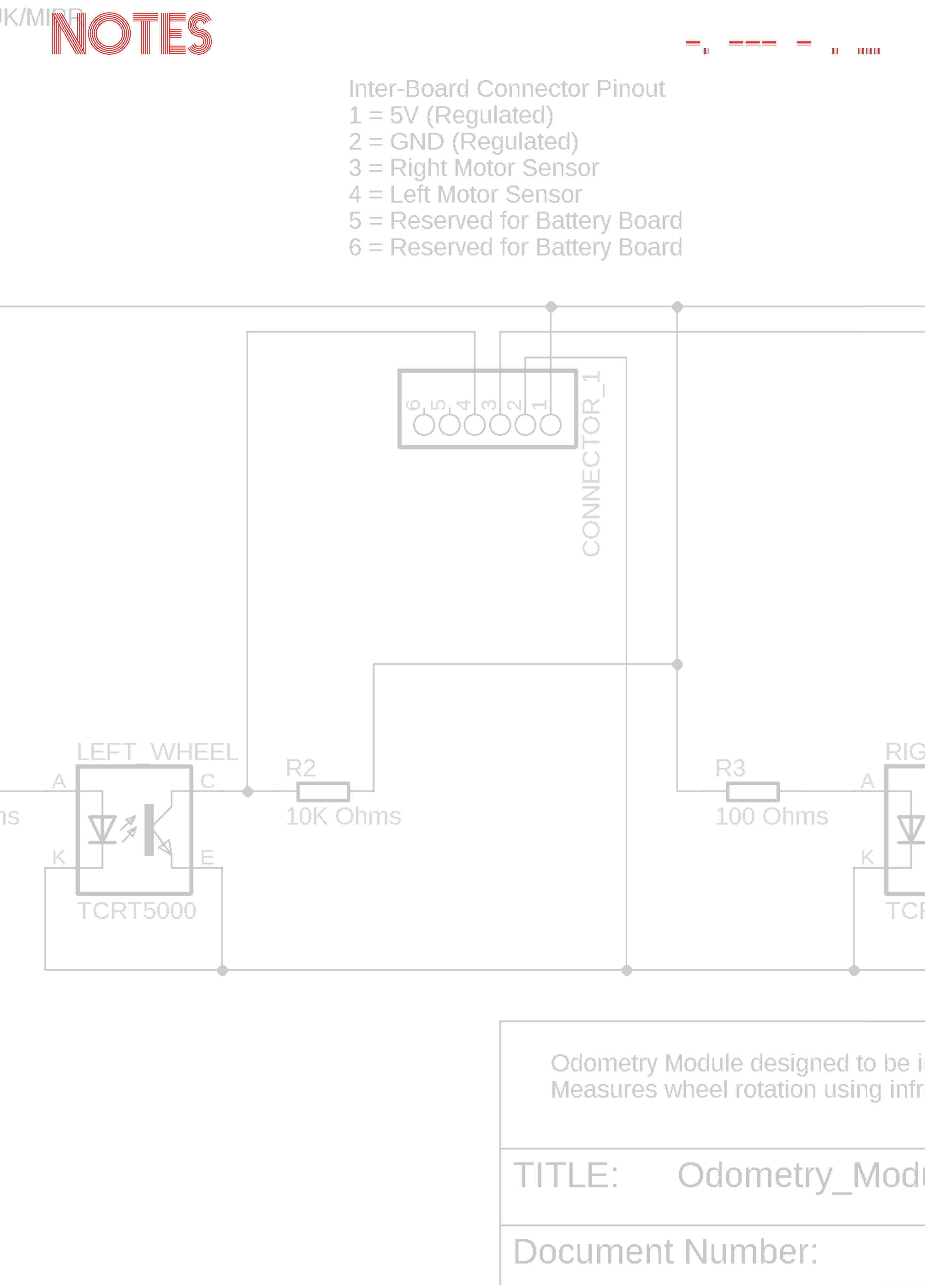
The Odometry Module is optional, it adds two sensors measuring wheel rotation and calculating their velocities in cm/s.

The below commands can be used with this module to independently get data. You can also use Telemetry Packets to get this data. For more information see the Telemetry section of this document.

If the loop time exceeds 4mS the Odometry Module will loose accuracy.

COMMAND	TYPE	DESCRIPTION	BOARD	EXAMPLE	RETURN VALUES
ENODO	Ordinary	Enables the Odometry Module	BATT/ODO	ENODO	Nil
DIENO	Ordinary	Disables the Odometry Module	BATT/ODO	DIENO	Nil
OLSPD	Ordinary	Gets the speed of the left when in cm/s	BATT/ODO	OLSPD	Integer value from 0 to maximum speed of around 50 cm/s of around 50cm/s
ORSPD	Ordinary	Gets the speed of the right wheel in cm/s	BATT/ODO	ORSPD	Integer value from 0 to maximum speed
BVOLT	Ordinary	Returns the battery voltage	BATT/ODO	BVOLT	Floating point number between 3.58 to 4.20

NOTES



Inter-Board Connector Pinout
1 = 5V (Regulated)
2 = GND (Regulated)
3 = Right Motor Sensor
4 = Left Motor Sensor
5 = Reserved for Battery Board
6 = Reserved for Battery Board

Odometry Module designed to be i
Measures wheel rotation using infr

TITLE: Odometry_Modu

Document Number:

TELEMETRY

Each parameter can be requested from MIPR separately however this is not always convenient, because of this you can use Telemetry Packets to receive sensor information. These packets are designed to give relevant sensor information for each sensor board regardless of whether the Odometry Module is attached.

The below table displays the commands that can be used. When MIPR is started in Op Mode 9 Telemetry is disabled and you must use the TELON command to enable it.

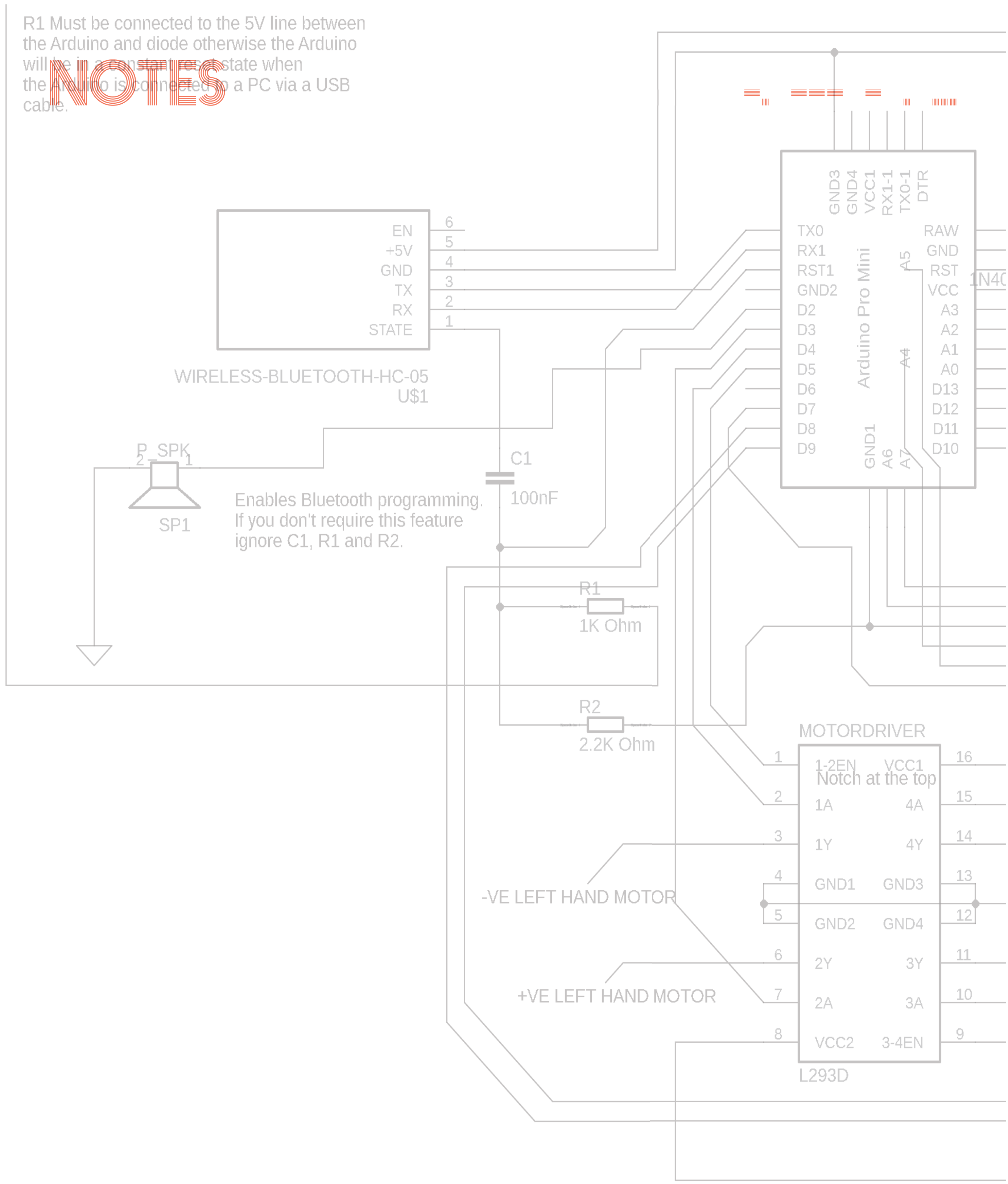
COMMAND	TYPE	DESCRIPTION	BOARD	EXAMPLE	RETURN VALUES
TELON	Ordinary	Turns the telemetry data on	N/A	TELON	Telemetry packets will be received
TELOFF	Ordinary	Turns the telemetry data off	N/A	TELOFF	Telemetry packets will stop
TEL	Command Value Pair	Sets the command packet	N/A	TEL:SB001T	You will receive telemetry packets for SB001 with the Odometry Module attached
REF	Command Value Pair	Sets the Telemetry Packet refresh rate in mS	N/A	REF:100	The telemetry packets will be displayed at the desired refresh rate

Telemetry Packets

The below table details the telemetry packets that can be used with the TEL command.

PACKET NAME	BOARD	EXAMPLE COMMAND	RETURN FORMAT
NILT	ODO	TEL:NILT	{LEFT VELOCITY, RIGHT VELOCITY, BATT VOLTAGE, LOOP TIME}
SB001F	SB001	TEL:SB001F	{LEFT LDR VAL, RIGHT LDR VAL, LOOP TIME}
SB001T	SB001 & ODO	TEL:SB001T	{LEFT LDR VAL, RIGHT LDR VAL, LEFT VELOCITY, LEFT MOTOR STATUS, RIGHT VELOCITY, RIGHT MOTOR STATUS, BATT VOLTAGE, LOOP TIME}
SB001AF	SB001A	TEL:SB001AF	{OBJECT DISTANCE, LEFT LDR VAL, RIGHT LDR VAL, LOOP TIME}
SB001AT	SB001A & ODO	TEL:SB001AT	{OBJECT DISTANCE, LEFT LDR VAL, RIGHT LDR VAL, LEFT VELOCITY, LEFT MOTOR STATUS, RIGHT VELOCITY, RIGHT MOTOR STATUS, BATT VOLTAGE, LOOP TIME}
SB002F	SB002	TEL:SB002F	{L1, L2, M, R2, R1, PID Input, PID Output, Left Right Error, LOOP TIME}
SB002T	SB002 & ODO	TEL:SB002T	{L1, L2, M, R2, R1, PID Input, PID Output, Left Right Error, LEFT VELOCITY, LEFT MOTOR STATUS, RIGHT VELOCITY, RIGHT MOTOR STATUS, BATT VOLTAGE, LOOP TIME}

NOTES



R1 Must be connected to the 5V line between the Arduino and diode otherwise the Arduino will be in a constant reset state when the Arduino is connected to a PC via a USB cable.

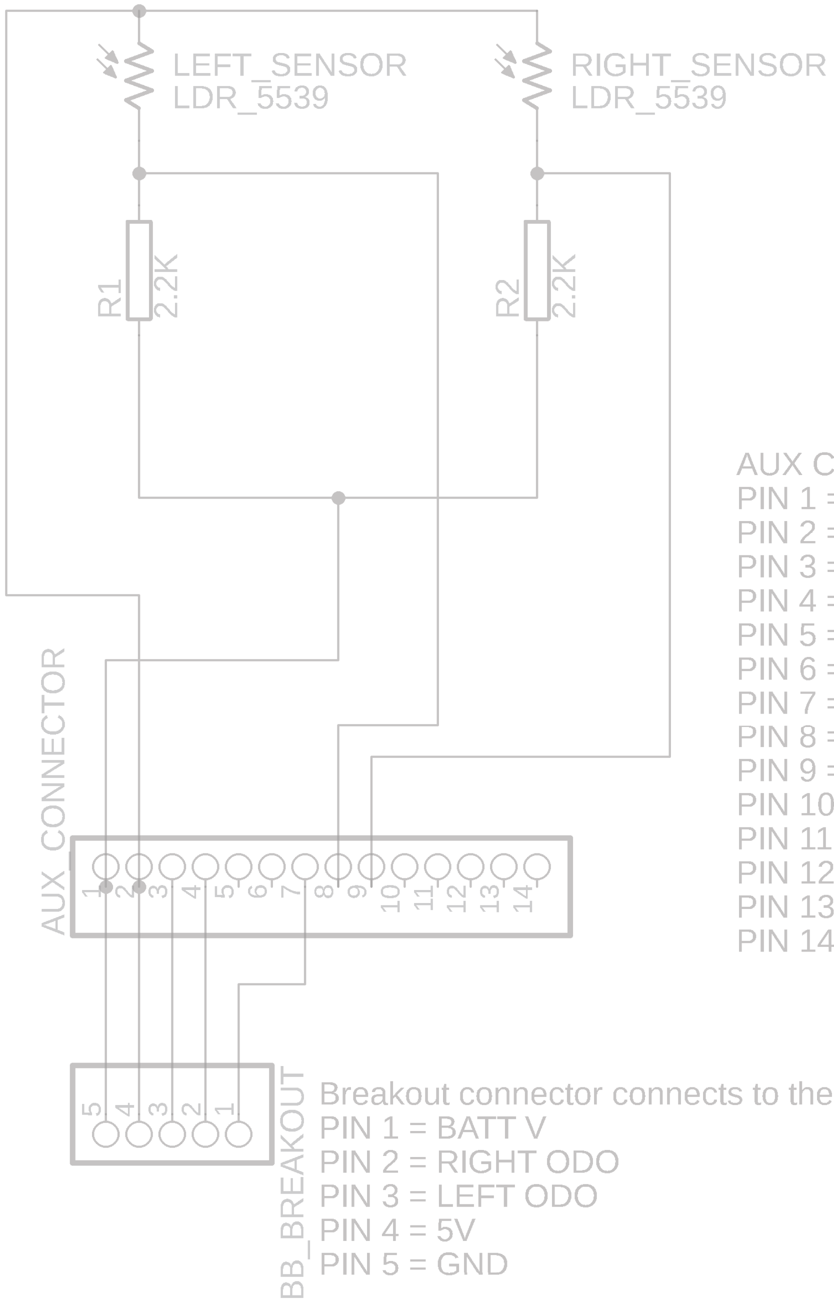
Enables Bluetooth programming. If you don't require this feature ignore C1, R1 and R2.

SB001A

These commands are to be used with SB001 and SB001A. They allow you to use the VL53L1X sensor and light dependent resistors. Please note that in order to use the VL53L1X it must be initialised using the SETSEN command. Telemetry Packets can also be used to get these values as well as the commands below.

COMMAND	TYPE	DESCRIPTION	BOARD	EXAMPLE	RETURN VALUES
MAXD	Command Value Pair	Sets the maximum range of the VL53L1X sensor. Values are in mm	SB001A	MAXD:4000	Nil
LDRL	Ordinary	Returns the value from the left light sensitive resistor	SB001/SB001A	LDRL	Integer Value from 0 to 1024
LDRR	Ordinary	Returns the value from the right light sensitive resistor	SB001/SB001A	LDRR	Integer Value from 0 to 1024
SETSEN	Ordinary	Initialises the VL53L1X sensor	SB001A	SETSEN	No error will return “SB001A Setup” An error will return “VL53L1X Fails to startup“
DIST	Ordinary	Reads and returns a distance value in mm from the VL53L1X sensor. SETSEN must be used before this command will return a value	SB001A	DIST	Returns an integer value between 0 and MAXD, MAXD is set at 2000 by default

NOTES



AUX Connector Pinout
PIN 1 = GND
PIN 2 = 5V
PIN 3 = LEFT ODO
PIN 4 = RIGHT ODO
PIN 5 = N/C
PIN 6 = N/C
PIN 7 = BATT V
PIN 8 = LEFT LDR
PIN 9 = RIGHT LDR
PIN 10 = N/C
PIN 11 = N/C
PIN 12 = N/C
PIN 13 = N/C
PIN 14 = N/C

Breakout connector connects to the battery board
PIN 1 = BATT V
PIN 2 = RIGHT ODO
PIN 3 = LEFT ODO
PIN 4 = 5V
PIN 5 = GND

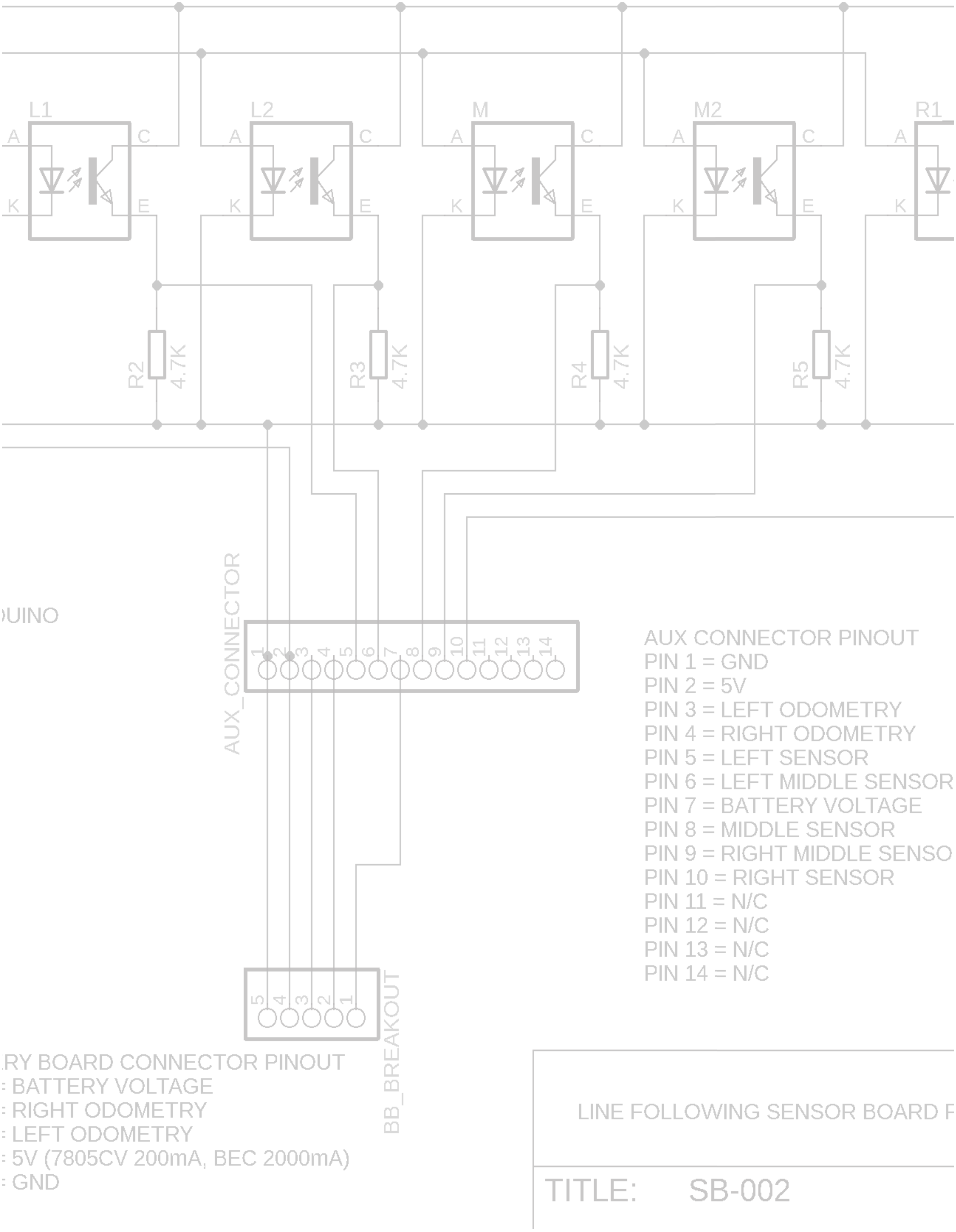
SB(
TITL
Docu
Date

SB002

These commands are to be used with SB002. This sensor board will need to be calibrated every-time it's switched on or when it's put on a new track as the reflectance properties could be different. These values can be returned using Telemetry Packets or the commands stated below.

COMMAND	TYPE	DESCRIPTION	BOARD	EXAMPLE	RETURN VALUES
ISCALD	Ordinary	Returns true or false depending if the sensors been calibrated	SB002	ISCALD	boolean values true or false
C	Ordinary	Calibrates the sensor, must be on the white part of the track and not over a black line	SB002	C	Nil - Calibration will take around 2 seconds
GET002BIAS	Ordinary	Returns bias values for a sensors and the basline value once the sensor is calibrated	SB002	GET002BIAS	Returns 5 values; {L1Bias, L2Bias, R2Bias, R1Bias, Basline}
GETCALDVAL	Ordinary	Returns values from all 5 sensors after calibration	SB002	GETCALDVAL	Returns 5 sensor reading values; {L1, L2, M, R2, R1}
GETUNCALDVAL	Ordinary	Returns values from all 5 sensors before calibration	SB002	GETUNCALDVAL	Returns 5 sensor reading values; {L1, L2, M, R2, R1}

NOTES



AUX CONNECTOR PINOUT
PIN 1 = GND
PIN 2 = 5V
PIN 3 = LEFT ODOMETRY
PIN 4 = RIGHT ODOMETRY
PIN 5 = LEFT SENSOR
PIN 6 = LEFT MIDDLE SENSOR
PIN 7 = BATTERY VOLTAGE
PIN 8 = MIDDLE SENSOR
PIN 9 = RIGHT MIDDLE SENSO
PIN 10 = RIGHT SENSOR
PIN 11 = N/C
PIN 12 = N/C
PIN 13 = N/C
PIN 14 = N/C

.RY BOARD CONNECTOR PINOUT
= BATTERY VOLTAGE
= RIGHT ODOMETRY
= LEFT ODOMETRY
= 5V (7805CV 200mA, BEC 2000mA)
= GND

LINE FOLLOWING SENSOR BOARD F

TITLE: SB-002

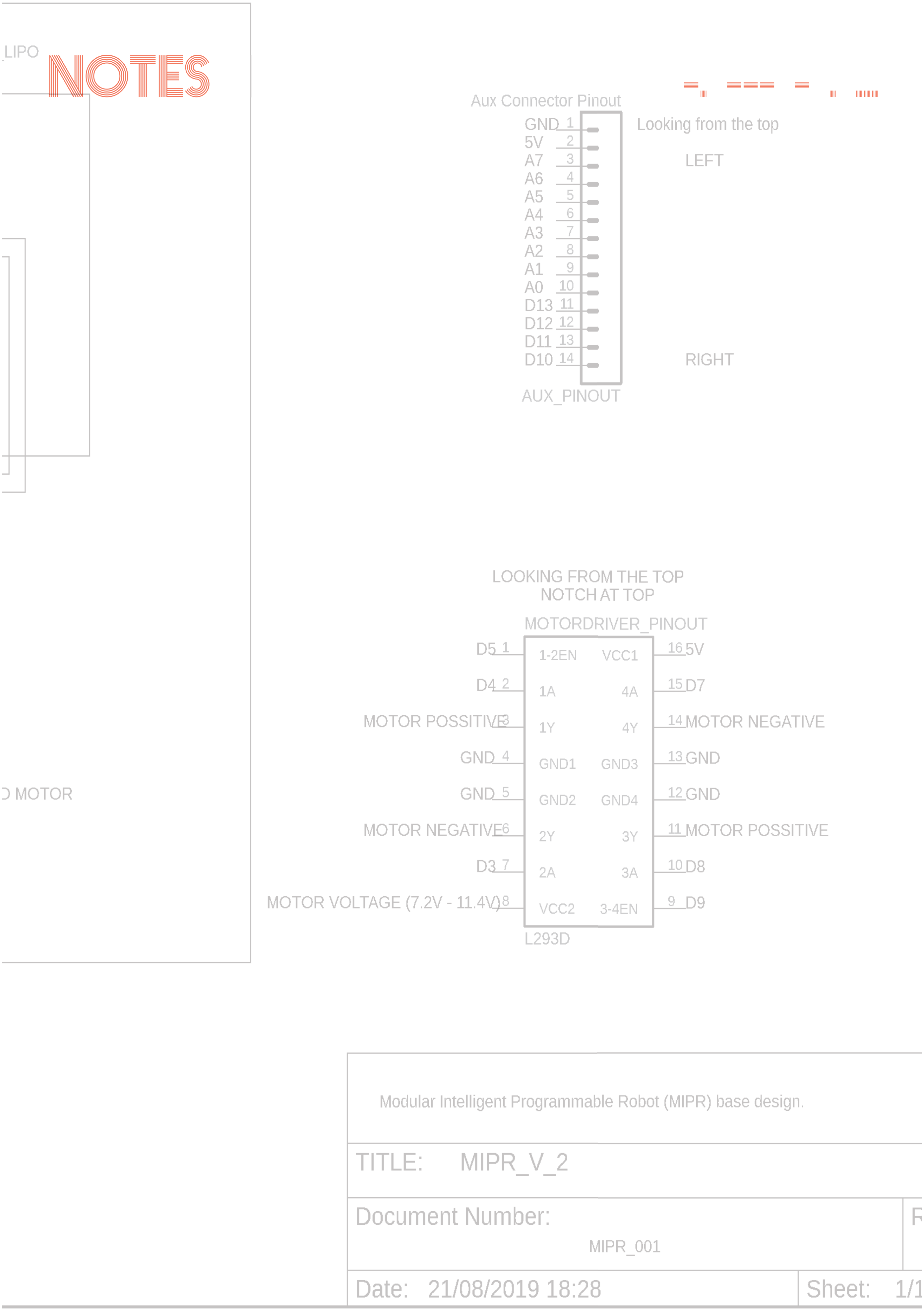
IO PINS

This SDK allows you to receive values from the IO pins as well as send signals to the pins. This will allow you to create your own sensor boards and quickly get values from analogue or digital sensors. Pins available are A0 to A5 (Odometry module installed) or A0 to A7 (without Odometry Module) and D10 to D13.

A4 and A5 are used for the I2C bus. If you are using this bus do not try to read from or write to these pins.

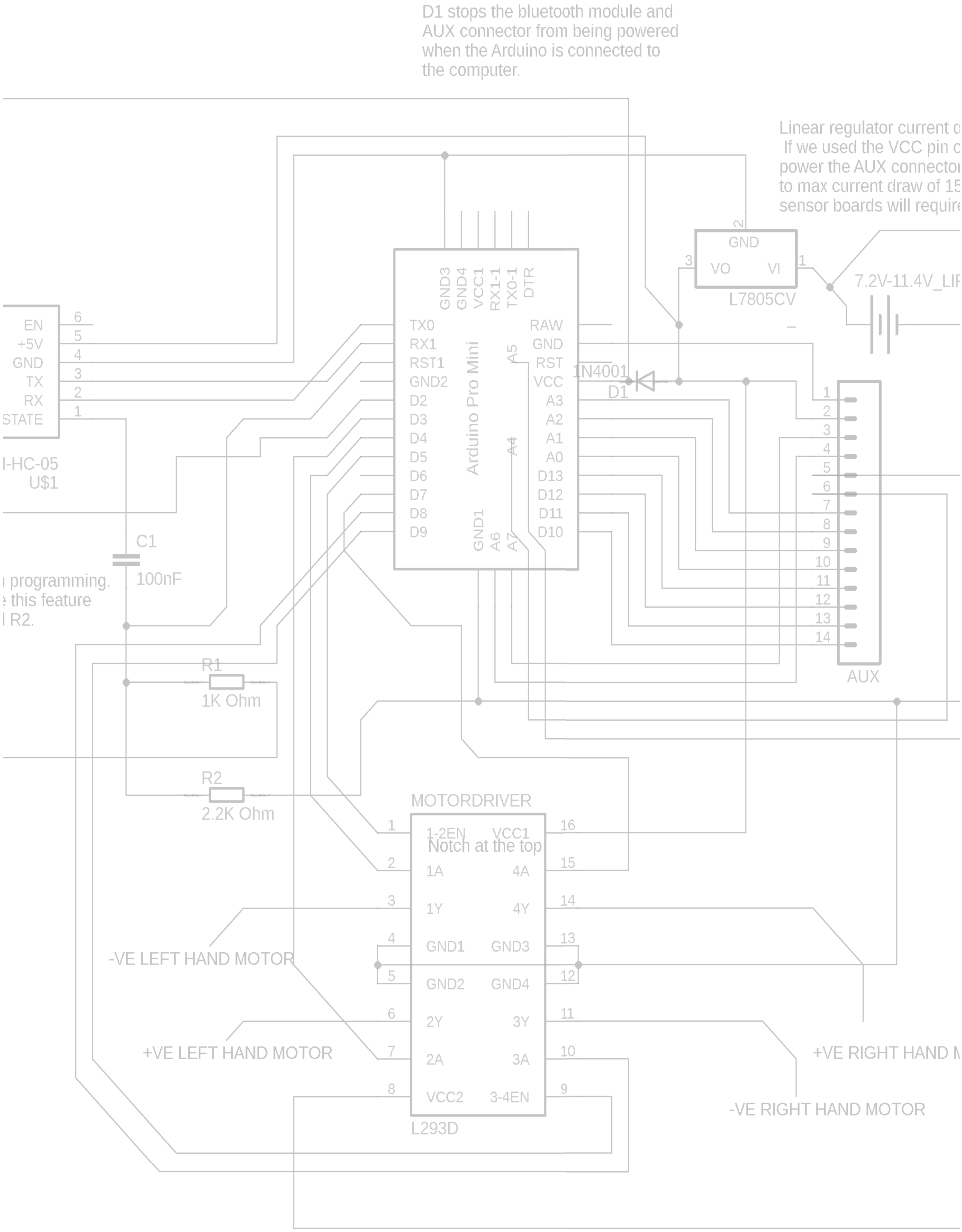
COMMAND	TYPE	DESCRIPTION	BOARD	EXAMPLE	RETURN VALUES
READPIN	Command Value Pair	Reads values from the selected pin	N/A	READPIN:A1 READPIN:D10	Integer value. 0 or 1 for digital pins or 0 to 1023 for analogue pins
WRITEPIN	Command Value Pair	Writes values to the selected pin	N/A	WRITEPIN:A1:255 WRITEPIN:D10:0	Nil. Values for digital pins will be 1 or 0. Values for analogue pins will be 0 to 1023

NOTES



LIPO
Aux Connector Pinout
GND 1
5V 2
A7 3
A6 4
A5 5
A4 6
A3 7
A2 8
A1 9
A0 10
D13 11
D12 12
D11 13
D10 14
Looking from the top
LEFT
RIGHT
AUX_PINOUT
D MOTOR
LOOKING FROM THE TOP
NOTCH AT TOP
MOTORDRIVER_PINOUT
D5 1 1-2EN
D4 2 1A
MOTOR POSSITIVE 3 1Y
GND 4 GND1
GND 5 GND2
MOTOR NEGATIVE 6 2Y
D3 7 2A
MOTOR VOLTAGE (7.2V - 11.4V) 8 VCC2
VCC1 16 5V
4A 15 D7
4Y 14 MOTOR NEGATIVE
GND3 13 GND
GND4 12 GND
3Y 11 MOTOR POSSITIVE
3A 10 D8
3-4EN 9 D9
L293D
Modular Intelligent Programmable Robot (MIPR) base design.
TITLE: MIPR_V_2
Document Number:
MIPR_001
R
Date: 21/08/2019 18:28
Sheet: 1/1

NOTES

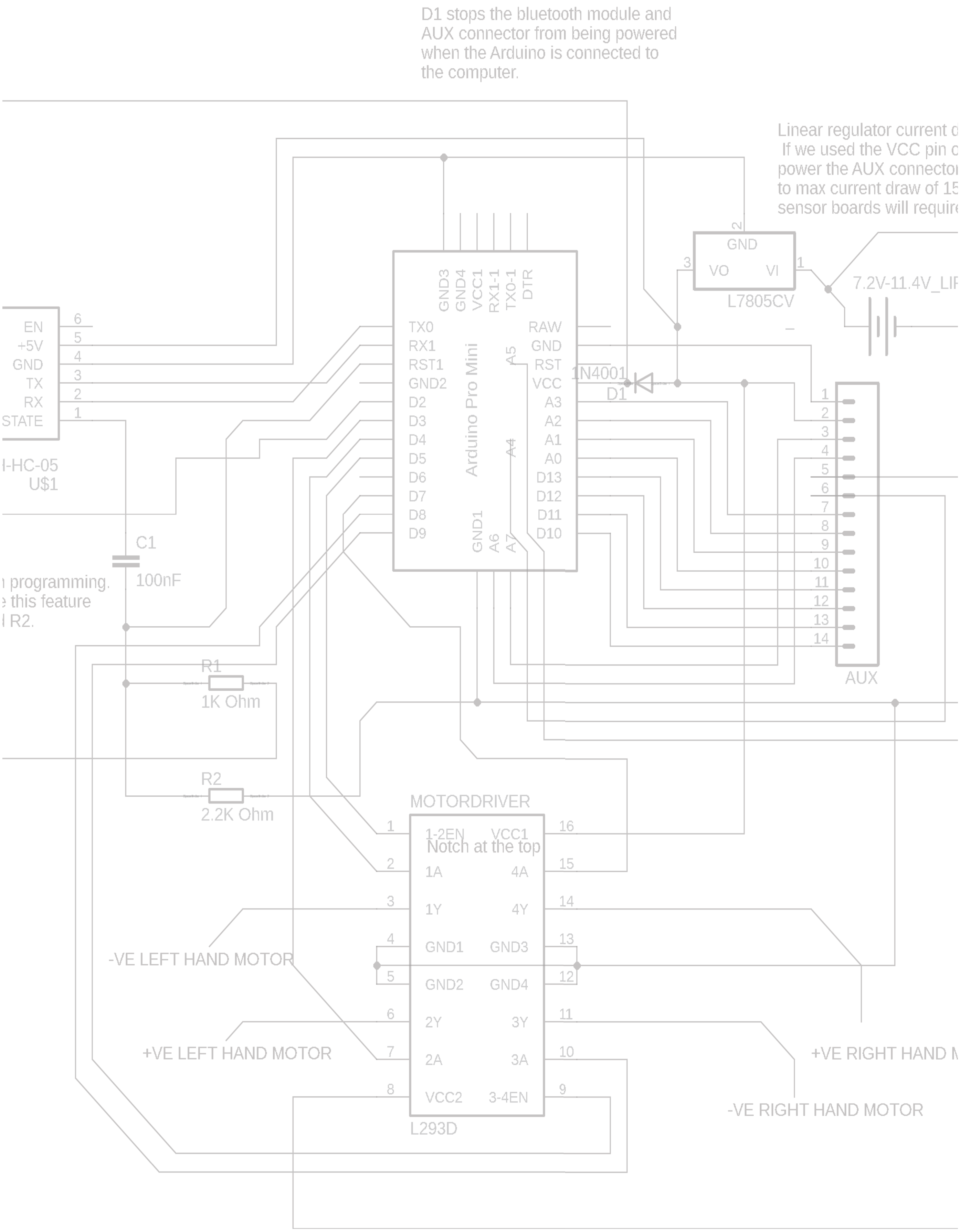


D1 stops the bluetooth module and AUX connector from being powered when the Arduino is connected to the computer.

Linear regulator current d
If we used the VCC pin o
power the AUX connector
to max current draw of 15
sensor boards will require

n programming.
e this feature
l R2.

NOTES



D1 stops the bluetooth module and AUX connector from being powered when the Arduino is connected to the computer.

Linear regulator current d
If we used the VCC pin o
power the AUX connector
to max current draw of 15
sensor boards will require

n programming.
e this feature
l R2.

-VE LEFT HAND MOTOR

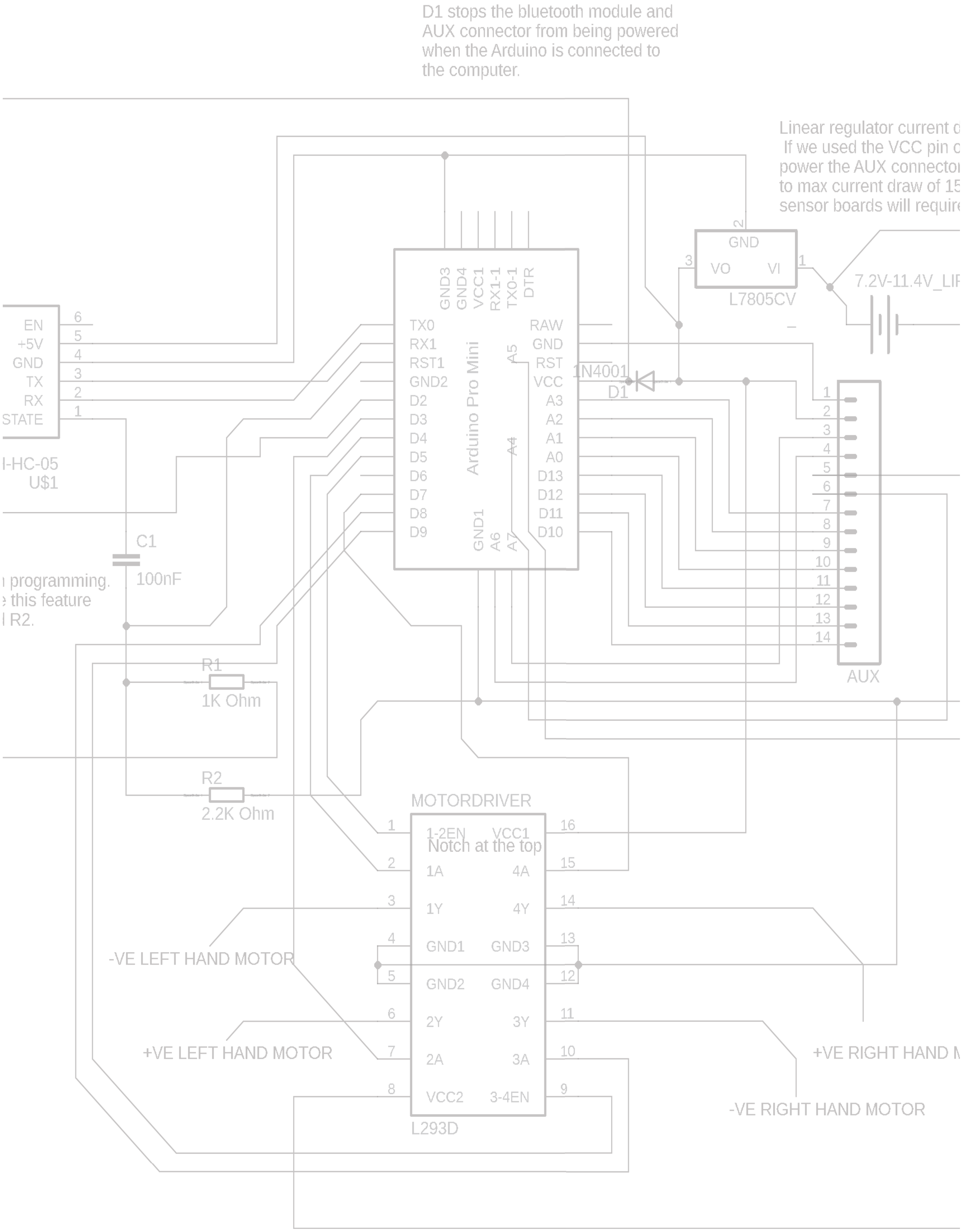
+VE LEFT HAND MOTOR

+VE RIGHT HAND M

-VE RIGHT HAND MOTOR

Notch at the top

NOTES



D1 stops the bluetooth module and AUX connector from being powered when the Arduino is connected to the computer.

Linear regulator current d
If we used the VCC pin o
power the AUX connector
to max current draw of 15
sensor boards will require

ı programming.
ꝫ this feature
l R2.

Notch at the top

L33T Components

HTTPS://L33T.UK/MIPR

HTTPS://L33T.UK/SDK