Simpelt program Dcc med delay

DCC Core Framework

Written by Philip Thomson @ mrlforum.co.uk (philip.thomson@f1p.uk)

Based on the DCC System by Tom Spink (tspink@gmail.com)

Compatible with Options:

1- H-Bridges using TIP120 & TIP125 Transistors (16V, 3A+) = Home & Pro Large Layouts

2- L298N H-Bridge using ENA, IN1, IN2 (12V 2A) = Very Small Layouts

3- LMD18200 H-Bridge using PWM, DIR, BRAKE (16V, 3A) = Reccomended most typical users

\*/

 /\*===================

 GLOBAL VARIABLES

 ===================\*/

 // Option 1 (TIP120 & TIP125) Variables

 /\*

 int A = 3; // Pin which controls A transistors

 int B = 4; // Pin which controls B transistors

 int speedCommand = 0x60;

 int lightingCommand = 0x9F;

 \*/

 // Option 2 (L298N) Variables

 int A = 3; // Pin which controls INPUT 1 on the L298N

 int B = 4; // Pin which controls INPUT 2 on the L298N

 int EN = 2; // Pin which controls ENABLE A on the L298N

 int speedCommand = 0x60;

 int lightingCommand = 0x9F;

 // Option 3 (LMD18200) Variables

 /\*

 int PWMP = 9; // Pin which controls PWM on the LMD18200

 int DIRP = 10; // Pin which controls DIR on the LMD18200

 int EN = 11; // Pin which controls BRAKE (Enable) on the LMD18200

 int speedCommand = 0x60;

 int lightingCommand = 0x9F;

 int instructionCommand;

 \*/

 // Current Sense Pin (For Short Circuit Protection)

 int CS = 5; // Pin connected to current sensor (e.g. ACS712)

 // Global Variables

 int fwd,rev,dccch; // Variables for Speed, Direction and Channel

// Setup Code - this only runs once

void setup() {

 pinMode(A, OUTPUT); // Define the Outputs from the Arduino

 pinMode(B, OUTPUT);

 pinMode(EN, OUTPUT);

 digitalWrite(EN, HIGH); // Enable the Output by Setting the chip HIGH

 digitalWrite(A, LOW); // Set Pins Low to start

 digitalWrite(B, LOW); // Set Pins Low to start

 dccch = 3; // Set Starting DCC Channel (Ch3 Default)

 delay(500);

}

// Looping Code

void loop() {

 fwd = 1;

 rev = 0;

 writeCmd(dccch, speedCommand);

 delayMicroseconds(10000);

 writeCmd(dccch, lightingCommand);

 delayMicroseconds(10000);

 writeCmd(dccch, speedCommand);

 if(fwd > 0)

 {

 speedCommand = speedCommand & ~0x20;

 speedCommand = (speedCommand & ~0x1F) | (fwd);

 }

 if(rev > 0)

 {

 speedCommand = speedCommand | 0x20;

 speedCommand = (speedCommand & ~0x1F) | (rev);

 }

 // Short Circuit Detection

 /\*

 if(analogRead(CS)>x){digitalWrite(EN, HIGH);}

 \*/

}

void write1()

 {

 digitalWrite(A, HIGH);

 digitalWrite(B, LOW);

 delayMicroseconds(48);

 digitalWrite(A, LOW);

 digitalWrite(B, HIGH);

 delayMicroseconds(48);

 }

 void write0()

 {

 digitalWrite(A, HIGH);

 digitalWrite(B, LOW);

 delayMicroseconds(96);

 digitalWrite(A, LOW);

 digitalWrite(B, HIGH);

 delayMicroseconds(96);

 }

 void writeByte(int b) {

 for (int i = 7; i >= 0; i--) {

 if ((b & (1 << i)) > 0) {

 write1();

 } else {

 write0();

 }

 }

 }

 void writeCmd(int addr, int data)

 {

 int checksum = addr ^ data;

 // Preamble

 for (int i = 0; i < 14; i++)

 write1();

 // Packet Start

 write0();

 writeByte(addr);

 // Data Start

 write0();

 writeByte(data);

 // Error Start

 write0();

 writeByte(checksum);

 // Packet End

 write1();

 write1();

 }