#include "simpletools.h"

#include "fdserial.h"

#define FALSE 0

#define TRUE !(FALSE)

// 1A 2A 3A 4A 5A 6A 1B 2B 3B 4B 5B 6B

unsigned int pin[12] = {6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17};

float servo\_angle[12];

int servo\_PWM[12];

// Walking Variables

float walk\_height = 90;

float lift\_height = 45;

float swing = 20;

float leg\_offset\_S[6] = {30, 0, -30, 30, 0, -30};

int swing\_sign\_SF[6] = {1, 1, -1, 1, -1, -1};

int swing\_sign\_SB[6] = {-1, -1, 1, -1, 1, 1};

float leg\_offset\_L[6] = {-30, 30, 0, -30, 30, 0};

int swing\_sign\_LF[6] = {-1, 1, 1, -1, 1, -1};

int swing\_sign\_LB[6] = {1, -1, -1, 1, -1, 1};

float leg\_offset\_R[6] = {0, -30, 30, 0, -30, 30};

int swing\_sign\_RF[6] = {1, -1, 1, -1, -1, 1};

int swing\_sign\_RB[6] = {-1, 1, -1, 1, 1, -1};

// Serial

fdserial \*ESP;

int sequence = 0;

// Forward Declarations

void reset\_servo\_angles(void);

void squat\_sequence(void);

void kill(void);

void walking\_sequence(int facing);

// Cog Routines & Stacks

void set\_servo\_1AB(void \*par);

void set\_servo\_2AB(void \*par);

void set\_servo\_3AB(void \*par);

void set\_servo\_4AB(void \*par);

void set\_servo\_5AB(void \*par);

void set\_servo\_6AB(void \*par);

unsigned int servo\_stack\_1[256];

unsigned int servo\_stack\_2[256];

unsigned int servo\_stack\_3[256];

unsigned int servo\_stack\_4[256];

unsigned int servo\_stack\_5[256];

unsigned int servo\_stack\_6[256];

// MAIN

int main(){

// Start Subroutines

cogstart(&set\_servo\_1AB, NULL, servo\_stack\_1, sizeof(servo\_stack\_1));

cogstart(&set\_servo\_2AB, NULL, servo\_stack\_2, sizeof(servo\_stack\_2));

cogstart(&set\_servo\_3AB, NULL, servo\_stack\_3, sizeof(servo\_stack\_3));

cogstart(&set\_servo\_4AB, NULL, servo\_stack\_4, sizeof(servo\_stack\_4));

cogstart(&set\_servo\_5AB, NULL, servo\_stack\_5, sizeof(servo\_stack\_5));

cogstart(&set\_servo\_6AB, NULL, servo\_stack\_6, sizeof(servo\_stack\_6));

// Reset Positions

reset\_servo\_angles();

// Game Loop

int T = 0;

int cap = 8;

pause(500);

while(T < cap){

walking\_sequence(1);

T++;

}

pause(350);

while(T < cap){

walking\_sequence(0);

T++;

}

pause(350);

while(T < cap){

squat\_sequence();

T++;

}

kill();

}

// Servo Sequences

void set\_leg\_position(int leg\_number, float theta\_1, float theta\_2){

servo\_angle[leg\_number] = theta\_2;

servo\_angle[leg\_number + 6] = theta\_1;

}

void reset\_servo\_angles(void){

for (int i = 0; i < 12; i++){

servo\_angle[i] = 90;

}

}

void walking\_sequence(int facing){

int t = 350;

float leg\_offset[6];

int swing\_sign[6];

switch (facing){

case 0:

for (int i = 0; i<6; i++){

leg\_offset[i] = leg\_offset\_L[i];

swing\_sign[i] = swing\_sign\_LF[i];

}

break;

case 1:

for (int i = 0; i<6; i++){

leg\_offset[i] = leg\_offset\_S[i];

swing\_sign[i] = swing\_sign\_SF[i];

}

break;

case 2:

for (int i = 0; i<6; i++){

leg\_offset[i] = leg\_offset\_R[i];

swing\_sign[i] = swing\_sign\_RF[i];

}

break;

case 3:

for (int i = 0; i<6; i++){

leg\_offset[i] = leg\_offset\_L[i];

swing\_sign[i] = swing\_sign\_LB[i];

}

break;

case 4:

for (int i = 0; i<6; i++){

leg\_offset[i] = leg\_offset\_S[i];

swing\_sign[i] = swing\_sign\_SB[i];

}

break;

case 5:

for (int i = 0; i<6; i++){

leg\_offset[i] = leg\_offset\_R[i];

swing\_sign[i] = swing\_sign\_RB[i];

}

break;

default:

for (int i = 0; i<6; i++){

leg\_offset[i] = leg\_offset\_S[i];

swing\_sign[i] = swing\_sign\_RF[i];

}

}

//

for (int i = 0; i < 3; i++){

int j = 2\*i;

set\_leg\_position(j, 90 - 0.5\*swing\_sign[j]\*swing + leg\_offset[j], lift\_height);

}

pause(t);

for (int i = 0; i < 3; i++){

int j = 2\*i + 1;

set\_leg\_position(j, 90 + leg\_offset[j], walk\_height);

}

pause(t);

//

for (int i = 0; i < 3; i++){

int j = 2\*i;

set\_leg\_position(j, 90 - swing\_sign[j]\*swing + leg\_offset[j], walk\_height);

}

pause(t);

for (int i = 0; i < 3; i++){

int j = 2\*i + 1;

set\_leg\_position(j, 90 - 0.5\*swing\_sign[j]\*swing + leg\_offset[j], lift\_height);

}

pause(t);

//

for (int i = 0; i < 3; i++){

int j = 2\*i;

set\_leg\_position(j, 90 + leg\_offset[j], walk\_height);

}

pause(t);

for (int i = 0; i < 3; i++){

int j = 2\*i + 1;

set\_leg\_position(j, 90 - swing\_sign[j]\*swing + leg\_offset[j], walk\_height);

}

pause(t);

}

void squat\_sequence(){

int t = 250;

pause(t);

for (int i = 0; i < 6; i++){

servo\_angle[i] = 120;

}

pause(t);

for (int i = 0; i < 6; i++){

servo\_angle[i] = 60;

}

}

void kill(){

while(TRUE){

for (int i = 0; i < 6; i++){

set\_leg\_position(i, 90, 50);

}

}

}

// Servo PWM subroutines

void set\_servo\_1AB(void \*par){

while(TRUE){

pulse\_out(pin[0], 400 + (190\*servo\_angle[0])/18);

pulse\_out(pin[6], 400 + (190\*servo\_angle[6])/18);

pause(20);

}

}

void set\_servo\_2AB(void \*par){

while(TRUE){

pulse\_out(pin[1], 400 + (190\*servo\_angle[1])/18);

pulse\_out(pin[7], 400 + (190\*servo\_angle[7])/18);

pause(20);

}

}

void set\_servo\_3AB(void \*par){

while(TRUE){

pulse\_out(pin[2], 400 + (190\*servo\_angle[2])/18);

pulse\_out(pin[8], 400 + (190\*servo\_angle[8])/18);

pause(20);

}

}

void set\_servo\_4AB(void \*par){

while(TRUE){

pulse\_out(pin[3], 400 + (190\*servo\_angle[3])/18);

pulse\_out(pin[9], 400 + (190\*servo\_angle[9])/18); // Opposite for some reason?

pause(20);

}

}

void set\_servo\_5AB(void \*par){

while(TRUE){

pulse\_out(pin[4], 400 + (190\*servo\_angle[4])/18);

pulse\_out(pin[10], 400 + (190\*servo\_angle[10])/18);

pause(20);

}

}

void set\_servo\_6AB(void \*par){

while(TRUE){

pulse\_out(pin[5], 400 + (190\*servo\_angle[5])/18);

pulse\_out(pin[11], 400 + (190\*servo\_angle[11])/18);

pause(20);

}

}