Programming Concepts C: GOTO_Beacon

ENGR 10
Introduction to Engineering
Main() for GOTO_Beacon Program

void main ( void )
{

freq=0; // 0= 1kHz (red), 1=10kHz (green beacon)
ambient_level=200; // used in 'move'
slow_level=5000; // used in 'move'
stop_level=6000; // used in 'move'
expose_time=3; // used in expose_and_react
steer_sensitivity=20; // used in 'move'
forward_speed=35; // forward speed, used in 'move'
slow_speed=25; // slow speed, used in 'move'
spin_speed=50; // spin speed (for searching mode), used in 'move'

SetDigitalOutput ( 14 , freq ); // turn to 1kHz (red beacon)

while ( 1==1 )
{
    Read_PD ( );
    find_max ( );
    move ( );
}

Note: you need to change freq for the beacon you want to find!

Global Variables

Infinite loop

Call three functions
Main() for GOTO_Beacon Program

Cortex

The only difference between the two programs are the port assignments

Move function

Motor ports 1 and 10

```c
void main ( void )
{
    freq=0; // 0=1khz (red), 1=10kHz(green beacon)
    ambient_level =200; // esed in 'move'
    slow_level =5000; // used in 'move'
    stop_level =6000; // used in 'move'
    expose_time =3; // used in expose_and_read
    steer_sensitivity =20; // used in 'move'
    forward_speed =35; // forward speed, used in 'move'
    slow_speed =25; // slow speed, used in 'move'
    spin_speed =50; // spin speed (for searching mode), used in 'move'
    SetDigitalOutput ( 12 , freq ); // turn to 1kHz (red beacon)
    while ( 1==1 )
    {
        Read_PD ( );
        find_max ( );
        move ( );
    }
    temp=limit_pwm(0+steer+speed);
    SetMotor ( 1 , temp ) ;
    temp=limit_pwm(0+steer-speed);
    SetMotor ( 10 , temp ) ;
}```
The main body of the GOTO_Beacon program is an infinite loop with three functions in the loop.

```c
spin_speed =50; // spin
SetDigitalOutput ( 12 ,
while ( 1==1 )
{
Read_PD (); //read 8 Photo-detector outputs
find_max (); //find the max detector output
move (); // move to the direction of max PD
}
```

PD4

PD7  Left side  Right side  PD0
Read_PD()

This function reads the intensity of all 8 Infrared Photo-detectors and stores the intensity (0 to 1023) into PD0~PD7 variables. It also produces the sum of all 8 readings (PD_sum). The sensing sensitivity is set by expose_time.

This function reads the intensity of either 1kHz or 10kHz infrared signal depending on the setting of digital output port 14, 12 on Cortex, (0=1kHz, 1=10kHz). You should set the digital output port #14 (#12 on Cortex) to the intended frequency before this function is called.

find_max()

This function determines which variable (among PD0~PD7) has the greatest value. The PD number is saved in the variable max_no.
move()

1. If PD_sum < Ambient (noise level, 200), the robot spins at the speed spin_speed (50). If PD_sum > Ambient (beacon is in view), go forward at speed of forward_speed (35).

2. Turns the left wheel faster if max_no < 4
   Turns the right wheel faster if max_no > 4
   Same speed if max_no = 4

3. Slows the forward motion to slow_speed if PD_sum > slow_level (5000)

4. Stop if PD_sum > stop_level (6000)
Clicker Question 1

Which of the following conditions indicates that the robot has reached the beacon?

(A) PD_Sum > Ambient
(B) PD_Sum > Stop_Level
(C) Speed < Slow_speed
(D) Speed == 0
(E) Spin_speed == 0
Clicker Question 2

If PD6 has the highest readout, what would the GOTO_Beacon program tell the robot to do?

(A) Make the robot spin
(B) Turn both motor at the same speed
(C) Makes the right motor run faster
(D) Makes the left motor run faster
(E) Stop the motors
Clicker Question 3

If PD_sum<Ambient, what would the GOTO_Beacon program do?

(A) Makes the robot spin
(B) Stop the motors
(C) Move at the **Slow_speed** level
(D) Makes the left motor run faster
(E) Makes the right motor run faster
```c
int Read_PD ( void )
{
    SetDigitalOutput ( 15 , 1 ); // close shutter, clear film, and increment counter
    SetDigitalOutput ( 16 , 1 ); // initialize counter value to '0'
    SetDigitalOutput ( 16 , 0 ); // allow counter to count
    Wait ( 5 ); // 5mS wait for things to settle down
    SetDigitalOutput ( 15 , 0 ); // open shutter and expose the film
    Wait ( expose_time ); // exposure time = 3mS to 8 mS
    PD0 = GetAnalogInput ( 1 ); // read intensity
    PD1 = expose_and_read ( );
    PD2 = expose_and_read ( );
    PD3 = expose_and_read ( );
    PD4 = expose_and_read ( );
    PD5 = expose_and_read ( );
    PD6 = expose_and_read ( );
    PD7 = expose_and_read ( );
    PD_sum=PD0+PD1+PD2+PD3+PD4+PD5+PD6+PD7
}
```
Main() for GOTO_Beacon Program

```c
void main ( void )
{

    freq=0; // 0= 1kHz (red), 1=10kz (green beacon

    ambient_level=200; // used in 'move'

    slow_level=5000; // used in 'move'

    stop_level=6000; // used in 'move'

    expose_time=3; // used in expose_and_reac

    steer_sensitivity=20; // used in 'move'

    forward_speed=35; // forward speed, used in 'move'

    slow_speed=25; // slow speed, used in 'move'

    spin_speed=50; // spin speed (for searching mode), used in 'move'

    SetDigitalOutput ( 14 , freq ); // turn to 1kHz (red beacon

    while ( 1==1 )
    {

        Read_PD ( ) ;

        find_max ( ) ;

        move ( ) ;

    }

    }

    }
# find_max() Function

```c
void find_max ( void )
{
    max_val=PD0;
    max_no=0;
    if ( PD1>max_val )
    {
        max_val=PD1;
        max_no=1;
    }
    if ( PD2>max_val )
    {
        max_val=PD2;
        max_no=2;
    }
    if ( PD3>max_val )
    {
        max_val=PD3;
        max_no=3;
    }
    if ( PD4>max_val )
    {
        max_val=PD4;
        max_no=4;
    }
    if ( PD5>max_val )
    {
        max_val=PD5;
        max_no=5;
    }
    if ( PD6>max_val )
    {
        max_val=PD6;
        max_no=6;
    }
    if ( PD7>max_val )
    {
        max_val=PD7;
        max_no=7;
    }
}
```
Clicker Question 4

The value of `expose_time` will affect the value of which of the following variable?

(A) Ambiant
(B) Slow_level
(C) Stop_level
(D) Spin_speed
(E) PD_sum
Main() for GOTO_Beacon Program

```c
void main ( void )
{

  freq=0;  // 0= 1kHz (red), 1=10kHz (green beacon)

  ambient_level=200;  // used in 'move'

  slow_level=5000;  // used in 'move'

  stop_level=6000;  // used in 'move'

  expose_time=3;  // used in expose_and_reac

  steer_sensitivity=20;  // used in 'move'

  forward_speed=35;  // forward speed, used in 'move'

  slow_speed=25;  // slow speed, used in 'move'

  spin_speed=50;  // spin speed (for searching mode), used in 'move'

  SetDigitalOutput ( 14 , freq );  // turn to 1kHz (red beacon)

  while ( 1==1 )
  {
    Read_PD ( );
    find_max ( );
    move ( );
  }
}
```

Move the robot
void move ( void )
{

    error=4-max_no; // heading direction error, if PD4 == max_no, then no error.
    steer=error*steer_sensitivity; // steering effort is proportional to heading error
    speed=forward_speed; // forward speed

    if ( PD_sum<ambient_level ) // If <background noise level => search mode
    {

        speed=0; // search mode => no forward motion
        steer=-spin_speed; // search mode => spin

    }

    if ( PD_sum>slow_level ) // Beacon is near!
    {

        speed=slow_speed; // Slow down

    }

    if ( PD_sum>stop_level ) // Found the beacon!
    {

        speed=0; // Stop
        steer=0; // no steering

    }

    temp=limit_pwm(127+steer+speed);
    SetPWM ( 2 , temp ) ;
    temp=limit_pwm(127+steer-speed);
    SetPWM ( 3 , temp ) ;
}

int limit_pwm ( int temp )
{
    if ( temp>255 )
    {
        limited=255;
    }
    else if ( temp<0 )
    {
        limited=0;
    }
    else
    {
        limited=temp;
    }
    return limited;
}
Example

PD2 reads max intensity \( \rightarrow \) max_no = 2 \( \rightarrow \) error = 4-2=2 \( \rightarrow \) steer=error*steer_sensitivity=2*20=40

Motor2 speed = 127+40+35=202, Motor3 speed=127+40-35=132, if motor 2 is on the left side the robot turns right

```c
void move ( void )
{
    error=4-max_no; // heading direction error, if PD4 == max_no, then no error.
    steer=error*steer_sensitivity; // steering effort is proportional to heading error
    speed=forward_speed; // forward speed

    temp=limit_pwm(127+steer+speed);
    SetPWM ( 2 , temp ) ;
    temp=limit_pwm(127+steer-speed);
    SetPWM ( 3 , temp ) ;
}
```
Clicker Question 6

Which of the following instructions or functions selects the target beacon (red or green)?

(A) \text{freq = 0;}
(B) \text{Stop\_level = 6000;}
(C) \text{Max\_PD()}
(D) \text{Setdigitaloutput(14,1)}
(E) \text{Move()}
How to add to the GOTO_beacon program to perform the required task?

Use **state variable**!

Initialization

While state==1

Read PD();
Find Max();
Move ();

While state==2

Try to turn off red beacon. when confirmed, state=3

While state==3

Go to green
When confirmed, state=4

Somewhere here, when red beacon is found, state=2.
No infinite loop in all blocks!
While state==4

While state==7

Capture green, When confirmed. state =5

GO home, When confirmed. state =6

STOP
IR Receiver Board

VEX Controller
- AD1
- AD14
- AD15
- AD16

Intensity output
Frequency select
Exposure control
Reset to detector #0

Exposure Control
Tuning circuit
amplifier

sensitivity

counter

reset

8
4
2
1

Selector

Infrared detectors

0
1
2
3
4
5
6
7

Counter

Sensitivity

Circuit board

Side view
Longer lead
Board edge
Top View

Circuit board
Side view
Longer lead
Board edge

21
Beacon Frequency Selection (filter)

amplifier

Tuning circuit

From previous stage

To next stage

1kHz

AD14
Freq select (0=1kHz, 1=10kHz)
IR Detector Selection

Infrared Detectors

LEDs

Selector

Counter

AD16 Counter reset

AD15 pulse

1 kHz

Increment count (0 to 1 transition)

1 Count output set to zero

0 Get ready to count

To next stage
Beacon Frequency Selection (filter)

From previous stage

1kHz

To next stage

amplifier

Tuning circuit

AD14
Freq select (0=1kHz, 1=10kHz)
Exposure Control Circuit

From previous stage
AD15

1. Clear ‘film’ (discharge C14)
2. Open ‘shutter’ (allow C14 to charge)

Amplifier

To AD1
(VEX Analog input)