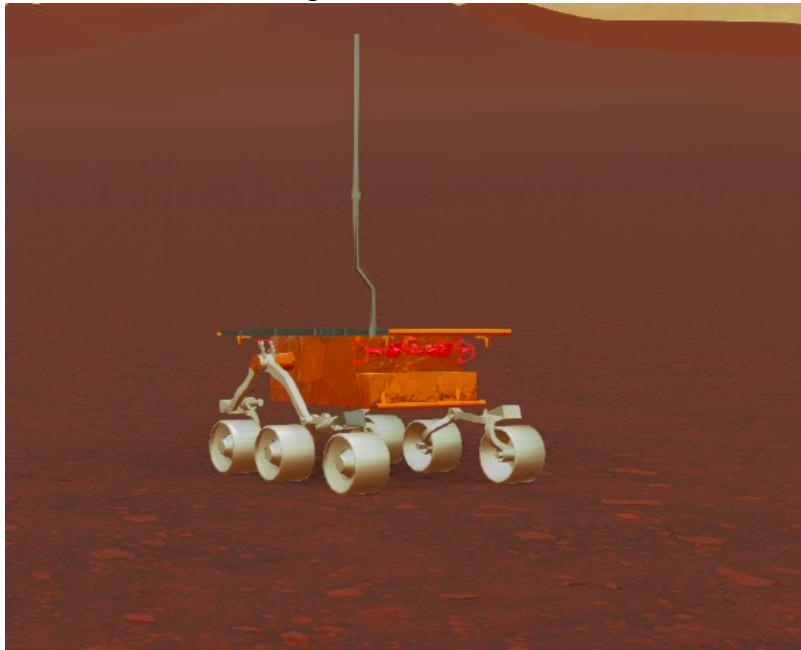


UJIAN TENGAH SEMESTER  
ROBOTIKA DAN SISTEM CERDAS

Nama : Muhamad Andi Darmawan  
Nim : 1103204182  
Kelas : TK-44-04

Hacking Webots : Nasa demo



Nama Robot: Nasa Sojourner, Tujuan : Robot ini merupakan robot yang dikembangkan nasa untuk melakukan eksplorasi di planet dan orbital planet seperti di Mars dan Bulan, Robot ini memiliki beragam fungsi dengan enam buah motor penggerak yang bisa melakukan gerakan maju, mundur, belok kanan kiri, mundur kiri-kanan dan berputar searah jarum jam untuk mengganti direksi

Berikut adalah source code robot Nasa Sojourner :

```
#include <stdio.h>
#include <webots/keyboard.h>
#include <webots/motor.h>
#include <webots/robot.h>

#define TIME_STEP 64
#define VELOCITY 0.6

enum {
```

```
back_left_bogie,  
front_left_bogie,  
front_left_arm,  
back_left_arm,  
front_left_wheel,  
middle_left_wheel,  
back_left_wheel,  
back_right_bogie,  
front_right_bogie,  
front_right_arm,  
back_right_arm,  
front_right_wheel,  
middle_right_wheel,  
back_right_wheel,  
JOINTS_MAX  
};
```

Program ini merupakan kode bahasa C yang menggunakan beberapa library dari simulator robot Webots. Library yang digunakan antara lain `stdio.h` untuk operasi input/output standar, `keyboard.h` untuk mengakses keyboard robot simulasi, `motor.h` untuk mengontrol motor robot, dan `robot.h` untuk menginisialisasi dan mengontrol robot itu sendiri.

Dalam program ini terdapat beberapa makro yang didefinisikan menggunakan `#define`. Makro pertama adalah `TIME_STEP` dengan nilai 64, yang digunakan untuk menentukan waktu simulasi. Makro kedua adalah `VELOCITY` dengan nilai 0.6, yang merupakan kecepatan robot dalam simulasi.

Selanjutnya, terdapat sebuah enum yang mendefinisikan tipe enumerasi dengan 14 konstanta bernama. Konstanta ini merepresentasikan sendi-sendi robot, seperti roda dan lengan robot. Setiap konstanta diberikan nilai integer yang dimulai dari 0 dan meningkat sebesar 1 untuk setiap konstanta berikutnya. Hal ini memungkinkan konstanta-konstanta ini digunakan sebagai indeks dalam array atau dalam switch statements.

```
WbDeviceTag joints[JOINTS_MAX];  
  
void move_4_wheels(double v) {  
    wb_motor_set_velocity(joints[front_left_wheel], v * VELOCITY);
```

```
wb_motor_set_velocity(joints[middle_left_wheel], v * VELOCITY);
wb_motor_set_velocity(joints[back_left_wheel], v * VELOCITY);
wb_motor_set_velocity(joints[front_right_wheel], v * VELOCITY);
wb_motor_set_velocity(joints[middle_right_wheel], v * VELOCITY);
wb_motor_set_velocity(joints[back_right_wheel], v * VELOCITY);

wb_motor_set_available_torque(joints[middle_right_wheel], 0.0);
wb_motor_set_available_torque(joints[middle_left_wheel], 0.0);
}
```

```
void move_6_wheels(double v) {
    wb_motor_set_available_torque(joints[middle_right_wheel], 2.0);
    wb_motor_set_available_torque(joints[middle_left_wheel], 2.0);

    wb_motor_set_velocity(joints[front_left_wheel], v * VELOCITY);
    wb_motor_set_velocity(joints[middle_left_wheel], v * VELOCITY);
    wb_motor_set_velocity(joints[back_left_wheel], v * VELOCITY);
    wb_motor_set_velocity(joints[front_right_wheel], v * VELOCITY);
    wb_motor_set_velocity(joints[middle_right_wheel], v * VELOCITY);
    wb_motor_set_velocity(joints[back_right_wheel], v * VELOCITY);
}
```

```
void turn_wheels_right() {
    wb_motor_set_position(joints[front_left_arm], 0.4);
    wb_motor_set_position(joints[front_right_arm], 0.227);
    wb_motor_set_position(joints[back_right_arm], -0.227);
    wb_motor_set_position(joints[back_left_arm], -0.4);
}
```

```
void turn_wheels_left() {
    wb_motor_set_position(joints[front_left_arm], -0.227);
    wb_motor_set_position(joints[front_right_arm], -0.4);
    wb_motor_set_position(joints[back_right_arm], 0.4);
    wb_motor_set_position(joints[back_left_arm], 0.227);
}
```

```
void wheels_straight() {
    wb_motor_set_position(joints[front_left_arm], 0.0);
    wb_motor_set_position(joints[front_right_arm], 0.0);
    wb_motor_set_position(joints[back_right_arm], 0.0);
}
```

```

    wb_motor_set_position(joints[back_left_arm], 0.0);
}

void turn_around(double v) {
    wb_motor_set_position(joints[front_left_arm], -0.87);
    wb_motor_set_position(joints[front_right_arm], 0.87);
    wb_motor_set_position(joints[back_right_arm], -0.87);
    wb_motor_set_position(joints[back_left_arm], 0.87);

    wb_motor_set_velocity(joints[front_left_wheel], v * VELOCITY);
    wb_motor_set_velocity(joints[middle_left_wheel], v * VELOCITY);
    wb_motor_set_velocity(joints[back_left_wheel], v * VELOCITY);
    wb_motor_set_velocity(joints[front_right_wheel], -v * VELOCITY);
    wb_motor_set_velocity(joints[middle_right_wheel], -v * VELOCITY);
    wb_motor_set_velocity(joints[back_right_wheel], -v * VELOCITY);

    wb_motor_set_available_torque(joints[middle_right_wheel], 0.0);
    wb_motor_set_available_torque(joints[middle_left_wheel], 0.0);
}

```

Pada program ini terdapat beberapa fungsi yang digunakan untuk mengontrol gerakan roda-roda pada robot, yaitu:

`move_4_wheels(double v)` : Fungsi ini digunakan untuk menggerakkan 4 roda di sisi kanan dan kiri robot dengan kecepatan  $v * VELOCITY$ . Fungsi ini juga mengatur available torque dari motor pada roda tengah agar bernilai 0.

`move_6_wheels(double v)` : Fungsi ini digunakan untuk menggerakkan 6 roda pada robot dengan kecepatan  $v * VELOCITY$ . Fungsi ini juga mengatur available torque dari motor pada roda tengah agar bernilai 2.

`turn_wheels_right()` : Fungsi ini digunakan untuk memutar roda-roda pada robot ke arah kanan dengan mengubah posisi motor pada lengan robot.

`turn_wheels_left()` : Fungsi ini digunakan untuk memutar roda-roda pada robot ke arah kiri dengan mengubah posisi motor pada lengan robot.

`wheels_straight()` : Fungsi ini digunakan untuk mengatur posisi motor pada lengan robot sehingga roda-roda pada robot bergerak secara lurus.

turn\_around(double v) : Fungsi ini digunakan untuk memutar robot 180 derajat dengan memutar roda-roda ke arah yang berlawanan dengan kecepatan  $v * VELOCITY$ . Fungsi ini juga mengatur available torque dari motor pada roda tengah agar bernilai 0.

```
int main() {
    // Required to initialize Webots
    wb_robot_init();

    joints[back_left_bogie] = wb_robot_get_device("BackLeftBogie");
    joints[front_left_bogie] = wb_robot_get_device("FrontLeftBogie");
    joints[front_left_arm] = wb_robot_get_device("FrontLeftArm");
    joints[back_left_arm] = wb_robot_get_device("BackLeftArm");
    joints[front_left_wheel] = wb_robot_get_device("FrontLeftWheel");
    joints[middle_left_wheel] = wb_robot_get_device("MiddleLeftWheel");
    joints[back_left_wheel] = wb_robot_get_device("BackLeftWheel");
    joints[back_right_bogie] = wb_robot_get_device("BackRightBogie");
    joints[front_right_bogie] = wb_robot_get_device("FrontRightBogie");
    joints[front_right_arm] = wb_robot_get_device("FrontRightArm");
    joints[back_right_arm] = wb_robot_get_device("BackRightArm");
    joints[front_right_wheel] = wb_robot_get_device("FrontRightWheel");
    joints[middle_right_wheel] = wb_robot_get_device("MiddleRightWheel");
    joints[back_right_wheel] = wb_robot_get_device("BackRightWheel");

    wb_motor_set_position(joints[front_left_wheel], INFINITY);
    wb_motor_set_position(joints[middle_left_wheel], INFINITY);
    wb_motor_set_position(joints[back_left_wheel], INFINITY);
    wb_motor_set_position(joints[front_right_wheel], INFINITY);
    wb_motor_set_position(joints[middle_right_wheel], INFINITY);
    wb_motor_set_position(joints[back_right_wheel], INFINITY);

    printf("Select the 3D window and use the keyboard to drive this robot:\n");
    printf("\n");
    printf("Q: forwards-left; W: forwards; E: forwards-right\n");
    printf("S: spin counter-clockwise\n");
    printf("Y: backwards-left; X: backwards; C: backwards-right\n");

    wb_keyboard_enable(TIME_STEP);
}
```

Pada baris pertama, dilakukan inisialisasi robot dengan memanggil fungsi `wb_robot_init()`.

Selanjutnya, terdapat pengaturan untuk setiap sendi atau motor pada robot dengan memanggil fungsi `wb_robot_get_device()` dan menyimpannya dalam array bernama `joints`. Terdapat 14 sendi yang diatur, yaitu `back_left_bogie`, `front_left_bogie`, `front_left_arm`, `back_left_arm`, `front_left_wheel`, `middle_left_wheel`, `back_left_wheel`, `back_right_bogie`, `front_right_bogie`, `front_right_arm`, `back_right_arm`, `front_right_wheel`, `middle_right_wheel`, dan `back_right_wheel`.

Kemudian, dilakukan pengaturan posisi motor menggunakan fungsi `wb_motor_set_position()` dengan nilai `INFINITY` pada roda-roda agar dapat berputar secara terus menerus.

Selanjutnya, dilakukan pencetakan pesan untuk memberi petunjuk penggunaan tombol pada keyboard untuk mengendalikan robot. Setelah itu, dilakukan aktivasi keyboard untuk memungkinkan pengguna untuk memberikan perintah pada robot dengan memanggil fungsi `wb_keyboard_enable()` dengan parameter `TIME_STEP`.

Secara keseluruhan, program ini memberikan dasar untuk mengendalikan robot dalam simulasi Webots menggunakan keyboard. Namun, tidak ada kode untuk membaca masukan pengguna atau untuk menggerakkan sendi pada robot. Oleh karena itu, perlu ditambahkan kode tambahan untuk memungkinkan robot bergerak dan merespons masukan pengguna.

```
// start moving
move_6_wheels(1.0);

while (wb_robot_step(TIME_STEP) != -1) {
    int key = wb_keyboard_get_key();
    switch (key) {
        case 'W':
            // forwards
            wheels_straight();
            move_6_wheels(1.0);
            break;
        case 'X':
            // backwards
            wheels_straight();
            move_6_wheels(-1.0);
            break;
        case 'Q':
            // forwards left
            turn_wheels_left();
            move_4_wheels(1.0);
            break;
```

```
case 'E':  
    // forwards right  
    turn_wheels_right();  
    move_4_wheels(1.0);  
    break;  
case 'Y':  
    // backwards left  
    turn_wheels_left();  
    move_4_wheels(-1.0);  
    break;  
case 'C':  
    // backwards right  
    turn_wheels_right();  
    move_4_wheels(-1.0);  
    break;  
case 'S':  
    // spin counter-clockwise  
    turn_around(1.0);  
    break;  
}  
}
```

program ini melakukan pengaturan motor pada setiap joint yang dimiliki robot. Kemudian, program menampilkan instruksi pada layar untuk menggerakkan robot dengan menggunakan keyboard.

Setelah itu, program memulai pergerakan pada keenam roda robot dengan memanggil fungsi `move_6_wheels(1.0)` untuk membuat robot bergerak maju. Selanjutnya, program akan memasuki loop `while` dan terus menerus membaca input dari keyboard menggunakan fungsi `wb_keyboard_get_key()`.

Jika input yang diterima adalah huruf 'W', maka fungsi `wheels_straight()` akan dipanggil untuk membuat roda robot bergerak lurus. Kemudian, fungsi `move_6_wheels(1.0)` dipanggil untuk membuat robot bergerak maju.

Jika input yang diterima adalah huruf 'X', maka fungsi `wheels_straight()` akan dipanggil untuk membuat roda-roda robot bergerak lurus. Kemudian, fungsi `move_6_wheels(-1.0)` dipanggil untuk membuat robot bergerak mundur.

Jika input yang diterima adalah huruf 'Q', maka fungsi `turn_wheels_left()` akan dipanggil untuk membuat roda-roda robot berbelok ke kiri. Kemudian, fungsi `move_4_wheels(1.0)` dipanggil untuk membuat robot bergerak maju ke arah kiri.

Jika input yang diterima adalah huruf 'E', maka fungsi `turn_wheels_right()` akan dipanggil untuk membuat roda-roda robot berbelok ke kanan. Kemudian, fungsi `move_4_wheels(1.0)` dipanggil untuk membuat robot bergerak maju ke arah kanan.

Jika input yang diterima adalah huruf 'Y', maka fungsi `turn_wheels_left()` akan dipanggil untuk membuat roda-roda robot berbelok ke kiri. Kemudian, fungsi `move_4_wheels(-1.0)` dipanggil untuk membuat robot bergerak mundur ke arah kiri.

Jika input yang diterima adalah huruf 'C', maka fungsi `turn_wheels_right()` akan dipanggil untuk membuat roda-roda robot berbelok ke kanan. Kemudian, fungsi `move_4_wheels(-1.0)` dipanggil untuk membuat robot bergerak mundur ke arah kanan.

Jika input yang diterima adalah huruf 'S', maka fungsi `turn_around(1.0)` akan dipanggil untuk membuat robot berputar searah jarum jam. Selama program berjalan, robot akan terus bergerak sesuai dengan input dari keyboard hingga program dihentikan secara manual.

```
wb_robot_cleanup();  
  
return 0;  
  
}
```

Setelah selesai menjalankan program, program melakukan pembersihan untuk memastikan bahwa sumber daya yang digunakan oleh Webots telah dibebaskan. Fungsi `wb_robot_cleanup()` digunakan untuk membersihkan semua sumber daya yang digunakan oleh simulasi Webots. Kemudian program mengembalikan nilai 0, yang menandakan bahwa program berakhir dengan sukses.