

## NQC Programs Explained

1)

```
task main()
{
    OnFwd(OUT_A + OUT_C);
    Wait(Random(1200));
    SetPower(OUT_B, 3);
    On(OUT_B);           // A lamp is connected to Port B.
    Wait(600);
    Toggle(OUT_C);
    Wait(40);
    Off(OUT_A + OUT_B + OUT_C);
}
```

Program starts with Motors A and C moving forward at power level 7 for a random amount of time (maximum random time of 12 seconds). Then a lamp in Port B turns on at power level 3 for 6 seconds. Motor A continues to go forward and the direction of motor C is toggled (flipped to reverse), both at full power, for .4 seconds, and then Ports A, B, and C stop.

2)

```
task main()
{
    repeat(3)
    {
        SetPower(OUT_A, 2);
        OnFwd(OUT_A);
        Wait(100);
        OnRev(OUT_A + OUT_C);
        Wait(200);
    }
    Off(OUT_A + OUT_C);
    repeat(4)
    {
        On(OUT_B);           // A lamp is connected to Port B.
        Wait(300);
        Off(OUT_B);
        Wait(100);
    }
}
```

Program starts with a motor in port A moving forward at power level 2 for 1 second. Then motors A and C move in reverse, A at power level 2 and C at power level 7, for 2 seconds. This entire task happens 3 times, and then Ports A and C are turned off. Then a lamp in port B comes on at power level 7 for 3 seconds. Power is cut off to port B for 1 second. This task (lamp B goes on for 3 seconds, then stops for 1 second) occurs a total of 4 times and the program ends.

3)

```
#define GO      300
#define STOP   200

task main()
{
    repeat(4)
    {
        OnFwd(OUT_A + OUT_C);
        Wait(GO);
        Off(OUT_A + OUT_C);
        Wait(STOP);
    }
    OnRev(OUT_A + OUT_C);
    Wait(250);
    Off(OUT_A + OUT_C);
}
```

Two constants, one named **GO** (value = 300), and another called **STOP** (value = 200) are defined. The following task is executed 4 times: Motors A and C move forward at power level 7 for 3 seconds (**GO**), and power is cut off to Ports A and C for 2 seconds (**STOP**). After the 4<sup>th</sup> loop, Motors A and C move in reverse at power level 7 for 2.5 seconds, and then motors A and C are stopped.

4)

```
int power;    int move;

task main()
{
    power = 7;
    move = 100;
    repeat(7)
    {
        SetPower(OUT_A + OUT_C, power);
        OnFwd(OUT_A + OUT_C);
        Wait(move);
        power -= 1;
        move += 20;
    }
    Off(OUT_A + OUT_C);
}
```

Two variables, one named **power**, and another named **move** are defined. **power** is given an initial value of 7, and **move** is given an initial value of 100. The following task is executed 7 times: Ports A and C are set to a power level equal to the value of the variable **power** (which is 7 for the 1<sup>st</sup> loop), Motors A and C move forward at that power level for a time that is equal to the value of the variable **move** (which is 100 for the 1<sup>st</sup> loop). At the end of each loop, the value of **power** decreases by 1 and the value of **move** increases by .2 seconds. This means that the final value of the variable **power** is 0 (although the motors never run at this power level) and the final value of the variable **move** is 240. After the 7<sup>th</sup> loop, Ports A and C are stopped.