

```

' {$STAMP BS2}
' {$PBASIC 2.5}

' 2007 mlab.taik.fi/paja
' Gimme Sugar v1.0

' (c) 2007 Jari Suominen, Anna Keune

' Version Log:
' v0.1
' This version just opens the valve completely and shuts it after
some amount of time.
' v0.2
' This version can pour in two different ways: full open with
sharp angle and half open
' with low angle.
' v0.3
' This version allready had a great led light show
' v0.4
' Optimized version
' v0.5
' Bugfixes and more optimization
' v0.6
' Cleaning up the Code
' v0.7
' Final tweaking of the parameters
' v1.0
' The perfect version

' ---[ I/O Definitions ]---
pLed1          PIN 3
pLed2          PIN 4
pLed3          PIN 5
pYin           PIN 9
pLightSensor  PIN 15
pMotor         PIN 12

' ---[ Variables ]---
wYPulse        VAR WORD      ' raw data from accelerometer
wYGravity      VAR WORD      ' Y axis gravity
wResult        VAR WORD

sugarInThePipe VAR WORD
pouring        VAR BIT

ledCounter     VAR WORD
ledState       VAR WORD
steps          VAR WORD

' ---[ Constants ]---
cPulseState    CON 1
cScale         CON $200          ' 2.0 us per unit

```

```

CENTER                CON  30000    'IS THIS ACTUALLY NEEDED? Yes it
is.
'Constants for the valve
FULLOPEN              CON  700
HALFOPEN              CON  580
CLOSED                CON  440

MAXSUGARINTHEPIPE    CON  50
FULLOPENSUGARCONSUMEDINITERATION  CON  5
HALFOPENSUGARCONSUMEDINITERATION  CON  2

ANGLEFORFULLOPEN     CON  65100
REFILLANGLE          CON  800

LEDSTATECHANGE       CON  50

' ---[ Initialization ]---
DEBUG CLS, "Gimme Sugar!", CR
ledState = 0
pouring = 0 'false
sugarInThePipe = 0
ledCounter = LEDSTATECHANGE + 1

' ---[ Main Code ]---
Main:
DO
    GOSUB Read_Accelerometer
    'DEBUG "y: ", DEC wYGravity, CR
    GOSUB Control_Valve
    GOSUB Update_Pouring_State
    GOSUB Do_The_Lightshow
LOOP
END

' ---[ Subroutines ]---
Control_Valve:
    IF sugarInThePipe = 0 THEN
        PULSOUT pMotor, CLOSED
    ELSEIF sugarInThePipe > 0 THEN
        IF pouring THEN
            IF wYGravity < ANGLEFORFULLOPEN THEN
                PULSOUT pMotor, FULLOPEN
                sugarInThePipe = sugarInThePipe -
FULLOPENSUGARCONSUMEDINITERATION
            ELSE
                PULSOUT pMotor, HALFOPEN
                sugarInThePipe = sugarInThePipe -
HALFOPENSUGARCONSUMEDINITERATION
            ENDIF
        IF sugarInThePipe > MAXSUGARINTHEPIPE THEN
            sugarInThePipe = 0

```

```

        ENDIF
        DEBUG "Sugar in the pipe: ", DEC sugarInThePipe, CR
    ELSE
        PULSOUT pMotor, CLOSED
    ENDIF
ENDIF
RETURN

```

Update_Pouring_State:

```

    IF wyGravity > CENTER THEN 'NOTE: negative numbers without
signs are big ones (> CENTER)
        IF pouring = 0 THEN
            IF sugarInThePipe > 0 THEN
                DEBUG "Gimme Sugar!", CR
                pouring = 1
            ELSE
                DEBUG "You have to refill!", CR
            ENDIF
        ENDIF
    ELSE ' wyGravity < CENTER
        IF pouring THEN
            pouring = 0
            DEBUG "Pouring ended!", CR
        ENDIF
        IF wyGravity > REFILLANGLE THEN
            IF sugarInThePipe < MAXSUGARINTHEPIPE THEN
                sugarInThePipe = MAXSUGARINTHEPIPE
                DEBUG "Refilled!", CR
            ENDIF
        ENDIF
    ENDIF
RETURN

```

Do_The_Lightshow:

```

    wResult = pLightSensor ' 0 or 1
    IF wResult = 1 THEN
        IF wyGravity < REFILLANGLE THEN
            ledCounter = ledCounter + 6*(REFILLANGLE-
wyGravity)/LEDSTATECHANGE
            'DEBUG DEC 6*(REFILLANGLE-wYGravity)/
LEDSTATECHANGE, CR
        ELSEIF wyGravity > CENTER THEN
            ledCounter = ledCounter + LEDSTATECHANGE + 1
            'forcing ledstate change on every iteration
        ENDIF
        IF sugarInThePipe = 0 THEN
            ledCounter = 0 'forcing ledstate to remain the same
in every iteration
        ENDIF
        IF ledCounter > LEDSTATECHANGE THEN
            ledState = ledState + 1
            IF ledState > 2 THEN

```

```

        ledState = 0
    ENDIF
    ledCounter = 0
ENDIF
IF ledState = 0 THEN
    HIGH pLed1
    LOW  pLed2
    LOW  pLed3
ELSEIF ledState = 1 THEN
    HIGH pLed2
    LOW  pLed1
    LOW  pLed3
ELSEIF ledState = 2 THEN
    HIGH pLed3
    LOW  pLed1
    LOW  pLed2
ENDIF
ELSE 'Shaker not in hand
    LOW  pled1
    LOW  pled2
    LOW  pled3
ENDIF
RETURN

Read_Accelerometer:
    PULSIN pYin, cPulseState, wYPulse
    wYPulse = wYPulse */ cScale
    wYGravity = ((wYPulse / 10) - 500) * 8
RETURN

```