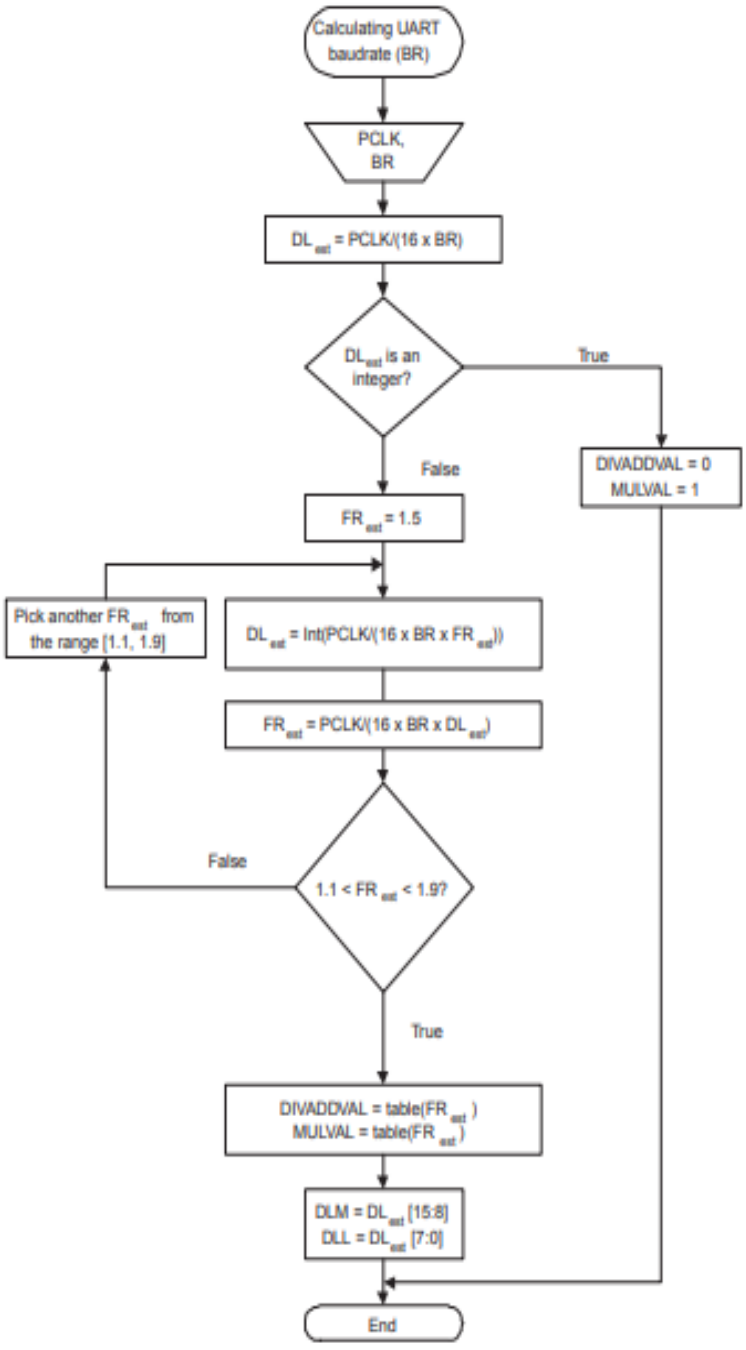

Fig 46. Algorithm for setting UART dividers

Python Program for setting the UART Dividers:

```
FR_LUT        = [1.000, 1.067,  1.071,  1.077,  1.083,  1.091,
1.100,  1.111,   1.125, 1.133,  1.143,  1.154,  1.167,  1.182,
1.200,  1.214,   1.222, 1.231,  1.267,  1.273,  1.286,  1.300,
1.308 , 1.333,   1.357, 1.364,  1.375,  1.385,  1.400,  1.417,
1.429,  1.444,   1.455, 1.462,  1.467,  1.533,  1.538,  1.545,
1.556,  1.571,   1.583, 1.600,  1.615,  1.625,  1.636,  1.643,
1.667,  1.692,   1.700, 1.714,  1.727,  1.733,  1.750 , 1.769,
1.778,  1.786,   1.800, 1.818,  1.833,  1.846,  1.857,  1.867,
1.875,  1.889,   1.900, 1.909,  1.917,  1.923,  1.929,  1.933]

DIVADDVAL_LUT = [ 0 , 1 ,     1 ,       1,        1,        1,
1,        1 ,       1,        2,        1,        2,        1,
2 ,       1 ,       3,        2,        3,        4,        3,
2,        3,        4,        1,        5,        4,        3,
5,        2,        5,        3,        4,        5,        6,
7 ,       8 ,       7,        6,        5,        4,        7,
3,        8,        5,        7 ,       9,        2,        9,
7,        5,        8,       11,        3,       10,        7,
11,       4,        9,        5,       11,        6,       13,
7,        8,        9,       10,       11,        12,       13,
14]

MULVAL_LUT    = [1,  15,    14,       13,       12,       11,
10,       9,        8,       15,        7,       13,        6,
11,       5,       14,        9,       13,       15,       11,
7 ,      10,       13,        3,       14,       11,        8,
13,       5,       12,        7,        9,       11,       13,
15,      15,       13,       11,        9,        7,       12,
5,       13,        8,       11,       14,        3,       13,
10,       7,       11,       15,       4,       13,        9,
14,       5,       11,        6,       13,        7,       15,
8,        9,       10,       11,       12,       13,       14,
15]
```

```
PCLK = eval(input("Enter the clock frequency:"))
BR = eval(input("Enter the Baudrate      :"))

DL_est = PCLK/(16*BR)
print("DL_est:",DL_est)

if DL_est.is_integer():
    DL_est=int(DL_est)
    print("in 1st if")
    print("DL_est",DL_est)
    DIVADDVAL=0
    MULVAL=1
```

```
    DLM = ((DL_est >> 8) & 0xFF)
    print("DLM   :", DLM)
    DLL = (DL_est & 0xFF)
    print("DLL   :", DLL)
else:
    print("in 1st else")
    FR_est = 1.5
    DL_est =int(PCLK/(16*BR*FR_est))
    print("DL_est:",DL_est)
    FR_est =(PCLK/(16*BR*DL_est))
    print("FR_est:", FR_est)
    range_val = len(FR_LUT)
    if(FR_est>1.1 and FR_est<1.9):
        print("in 2nd if")
        for i in range(0,range_val+1):
            if(FR_est>=FR_LUT[i] and FR_est<=FR_LUT[i+1]):
                DIVADDVAL = DIVADDVAL_LUT[i];
                MULVAL    = MULVAL_LUT[i];
                break;
        print("FR_est   :",FR_est)
        print("DIVADDVAL:", DIVADDVAL)
        print("MULVAL   :", MULVAL)
        DLM = ((DL_est >> 8) & 0xFF)
        print("DLM   :", DLM)
        DLL = (DL_est & 0xFF)
        print("DLL   :", DLL)
    else:
        print("in 2nd else")
        FR_est = 1.5
        DL_est = int(PCLK / (16 * BR * FR_est))
        for i in range(0,range_val+1):
            if(FR_est>=FR_LUT[i] and FR_est<=FR_LUT[i+1]):
                DIVADDVAL = DIVADDVAL_LUT[i];
                MULVAL    = MULVAL_LUT[i];
                break;
        print("DIVADDVAL:",DIVADDVAL)
        print("MULVAL   :",MULVAL)
        DLM = ((DL_est>>8)&0xFF)
        print("DLM   :", DLM)
        DLL = (DL_est&0xFF)
        print("DLL   :", DLL)


BAUDRATE_VAL=(PCLK/((16*(256*DLM+DLL))*(1+(DIVADDVAL/MULVAL))) )
print("BAUDRATE:",BAUDRATE_VAL)
```