DOE/CE/15482--T5

DOE/CE/15482--T5

DE92 005294

TECHNICAL PROGRESS REPORT

FOURTH QUARTER REPORT-1991

THE DEVELOPMENT AND TESTING OF A FIELDWORTHY SYSTEM OF IMPROVED FLUID PUMPING DEVICE AND LIQUID SENSOR FOR OIL WELLS

DEPARTMENT OF ENERGY GRANT # DE-FG01-90CE 15482

Principal Investigator: William G. Buckman, Ph. D.
504 Memphis Junction Road
Bowling Green, KY 42101

December 31, 1991

During the fourth quarter, we have concentrated on:

I. A MICROPROCESSOR BASED REMOTE OIL FIELD MONITOR AND CONTROLLER. This system has been designed to remotely monitor and control the pumping systems on oil and gas leases from a central location (office or home). The attached report is a detailed description of the design and development of this system.

II. SOLAR POWERED AIR PULSE OIL PUMP

We have also designed a system which will enable the Air Pulse Oil Pump System to operate using solar power for the pumping of low production shallow oil wells. This eliminates the necessity of running electrical lines to remote areas or the necessity of refueling gas powered pumps. A typical design of a solar powered shallow oil well producing system is attached. It is obvious that this system can also be used to pump water from reservoirs from depths as great as 500 feet.

MICROPROCESSOR BASED REMOTE OIL FIELD MONITOR AND CONTROLLER

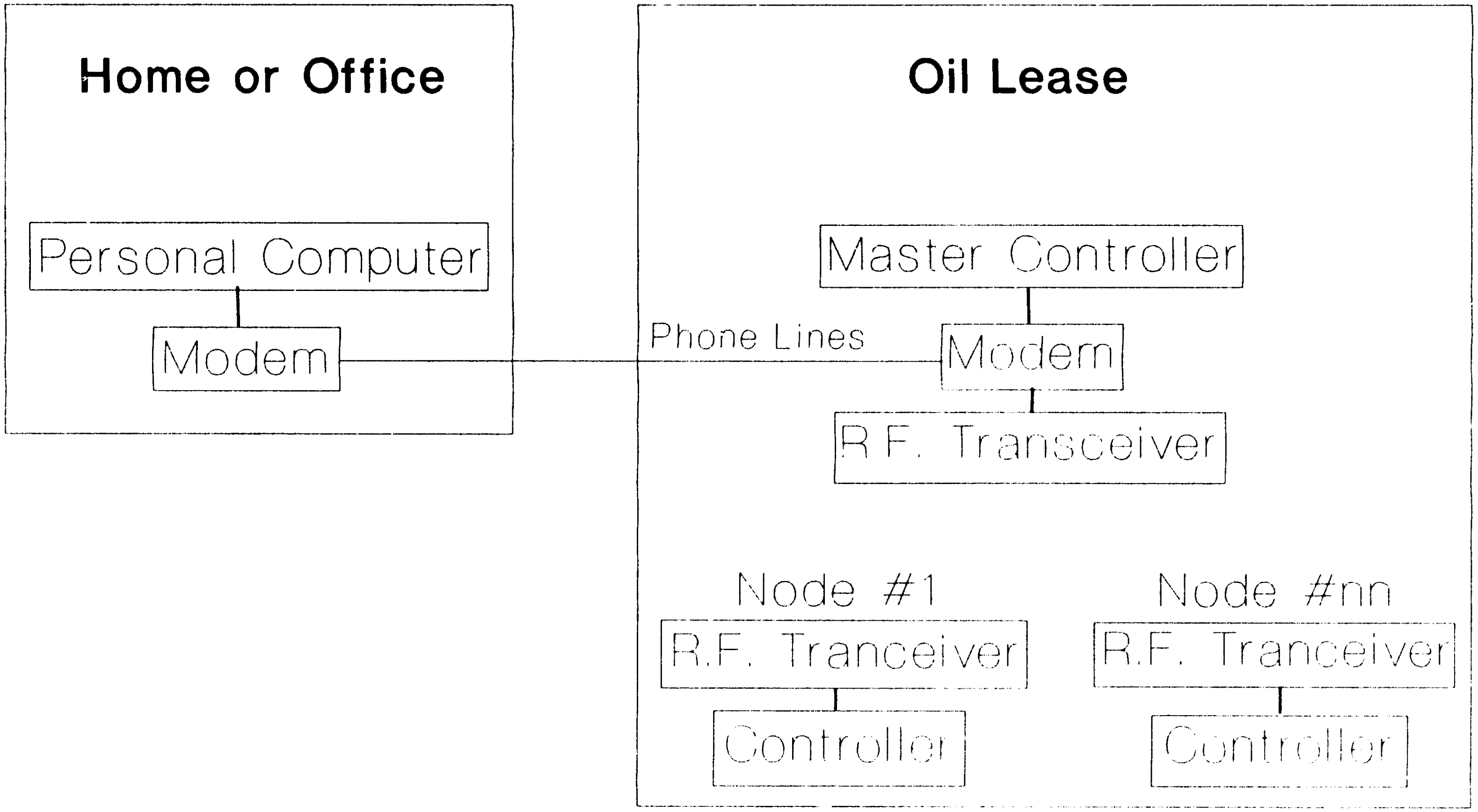
INTRODUCTION

A major expenditure to maintain oil and gas leases is the support of pumpers, those individuals who maintain the pumping systems on wells to achieve optimum production. Many leases are marginal and are in remote areas and this requires considerable driving time for the pumper. The Air Pulse Oil Pump System is designed to be an economical system for the shallow stripper wells. To improve on the economics of this system, we have designed a Remote Oil Field Monitor and Controller to enable us to acquire data from the lease to our central office at anytime and to control the pumping activities from the central office by using a personal computer. The advent and economics of low-power microcontrollers have made it feasible to use this type of system for numerous remote control systems. We can also adapt this economical system to monitor and control the production of gas wells and/or pump jacks.

OVERVIEW

While this system was designed to control the Air Pulse Oil Pump, it also allows an individual to remotely receive and transmit data to and from an oil lease via phone lines to acquire desired information concerning the production and security and to program the system to optimize the production of gas and/or oil leases. To achieve optimum production from a well, the system maintains the desired liquid level in the borehole by using the signal from a liquid level sensor in the borehole. To increase the overall efficiency of the Air Pulse Oil Pump, the pump cycle is automatically terminated when the oil slug is detected at the surface. The system will also react appropriately to low air line pressure which is another Air Pulse Oil Pump System concern. As shown in Figure 1. a personal computer can be used to initiate the data link between your home or office and the oil lease via modems and the phone lines or other modes of communication. Once the personal computer has access to the oil lease master controller, a radio frequency data link is established between the home computer and the well-head(node) controller via the master controller. At this point, the master controller becomes a relay of information and the user has complete control over the well-head controller. The well-head controller can be prompted to send statistical information to the home computer data base. This information consists of data such as: sleep time, sample time, pump time, exhaust time, number of pumps, number of pump terminations, number of samples, number of inquiries, etc. The user can also change any of the settings mentioned.

Figure 1.
Remote Microprocessor Controlled Pumping System



DESCRIPTION AND OPERATION

A block diagram of the well-head controller for controlling the pumping system is illustrated in Figure 2. There are presently four modes of operation of this system and it operates only in one mode at a time. These modes are named as follows: sleeping, sampling, pumping and exhausting. The microcontroller chip, MC68HC705C8, is responsible for sequencing and timing of these modes. During the sleep mode, the microcontroller deenergizes the unessential elements of the system and monitors the radio frequency channel for possible instructions. During the sample mode, the microcontroller applies power to the thermistor sensor in the borehole via a constant current source and monitors the thermistor voltage via a voltage comparator. When a constant current is applied to the negative temperature coefficient thermistor, it will begin to heat. The thermistor will heat much more rapidly in air as opposed to oil or water causing its resistance to be much lower. Since a constant current is being applied to the thermistor, the voltage across the thermistor(in air) will also be much lower. The voltage comparator compares the thermistor voltage to a known reference and sends a digital result(1 or 0) to the microcontroller. After a preset sample time interval, the microprocessor inspects the comparators output and either begins a pumping cycle or goes to sleep. During a pumping cycle, a relay is energized and other nonessentials are deenergized. The pump cycle can be terminated after a preset time or by a slug sensor when the liquid slug reaches the surface in the liquid discharge tube. The exhaust cycle is a delay to allow depressurization of the pumping system components and tubing down-hole and the refilling of the pump chamber and the tubing.

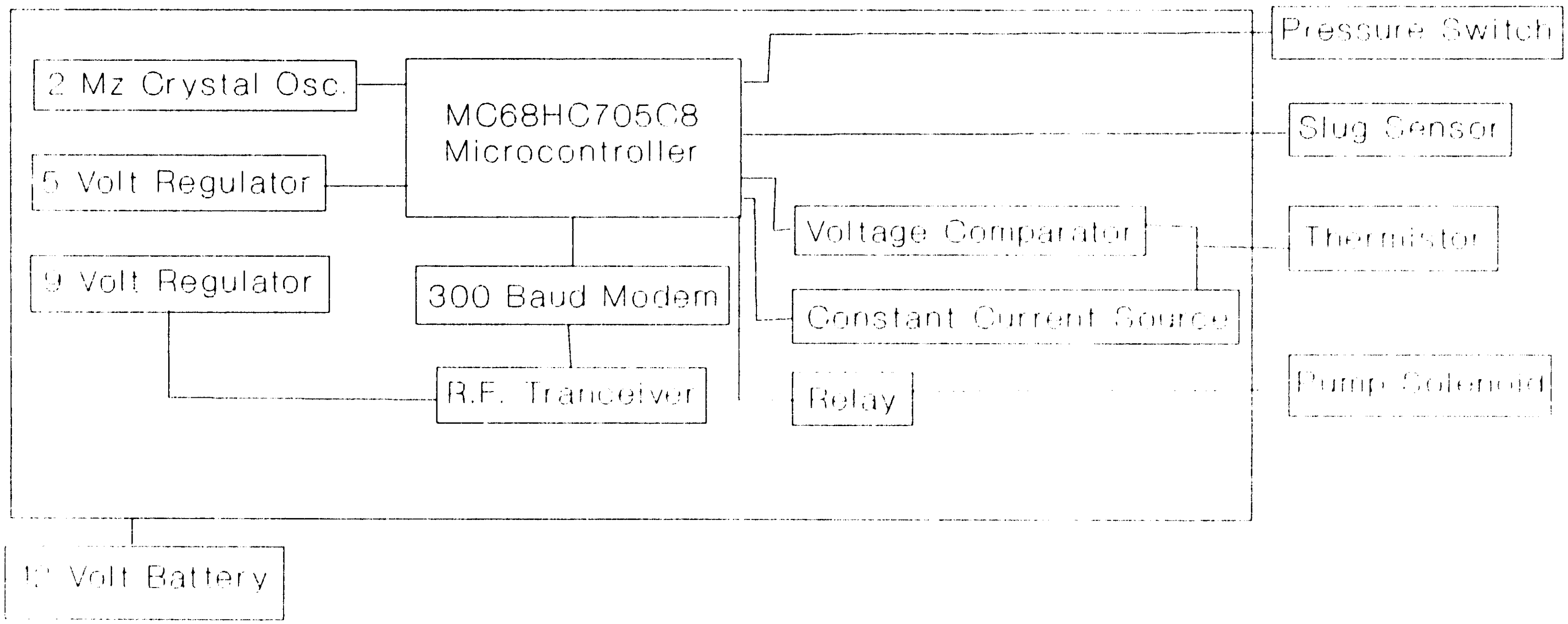
A radio frequency serial data link can be established to setup and control the system at a 300 baud rate of transmission via a modem and 49 Mz low-power radio frequency transceiver.

METHOD OF PROGRAMMING

The MC68HC705C8 microcontroller performs all of the control functions of this system. The microcontroller can be purchased as a one-time programmable or erasable/programmable chip. During the prototype stage, an erasable/programmable chip is used. Erasing is accomplished by exposure to ultraviolet light for approximately 5 minutes. Programming and debugging is accomplished via a Motorola evaluation module. The evaluation module accepts compiled programs from a personal computer and emulates the chip for debugging purposes. To test the target system(Well-Head Controller) a cable is connected from the evaluation module to a 40 pin dip microcontroller socket on the target system. After the compiled program is loaded into the evaluation module's memory, the evaluation module can be instructed to execute the program and emulate the microcontroller. Once the program is debugged, a

Figure 2.

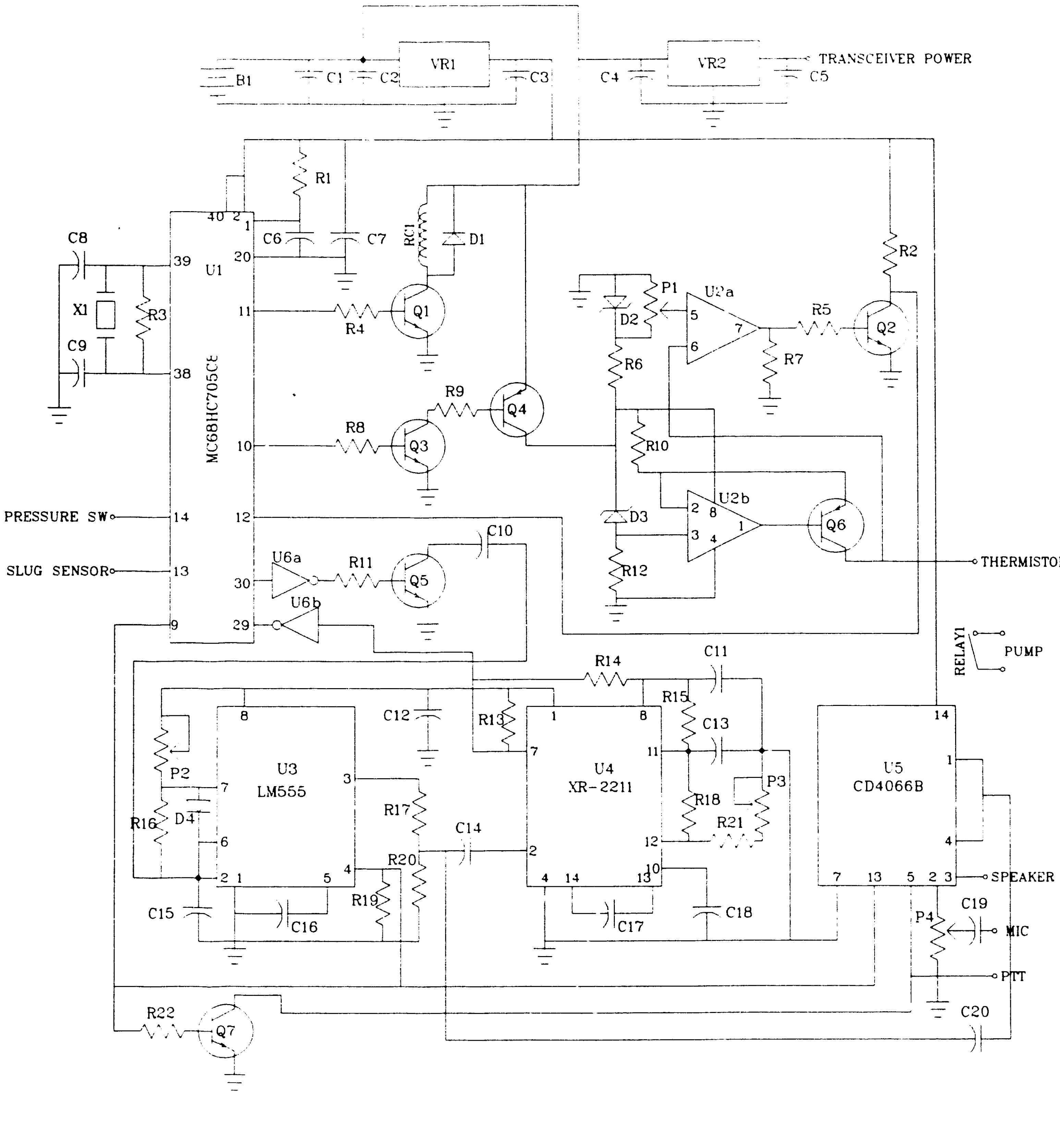
Remote Microprocessor Controlled Pumping System

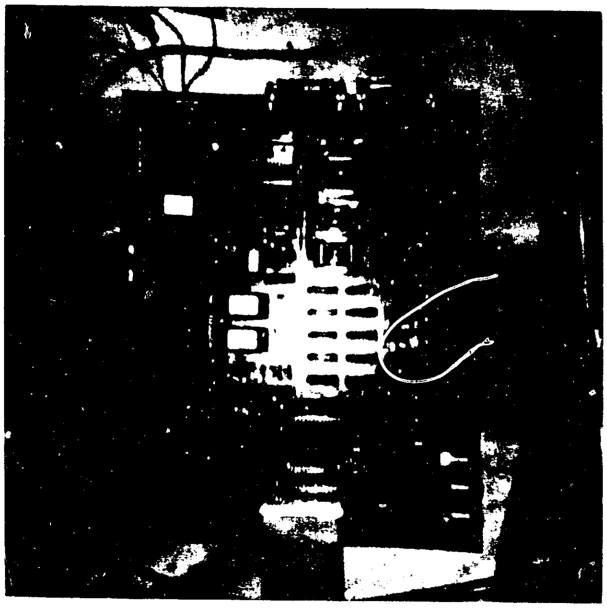


microcontroller chip is plugged into the evaluation module and the program is "burned-in". Finally, the cable from the evaluation module to the target system is disconnected and the microcontroller chip is inserted in its place on the target system.

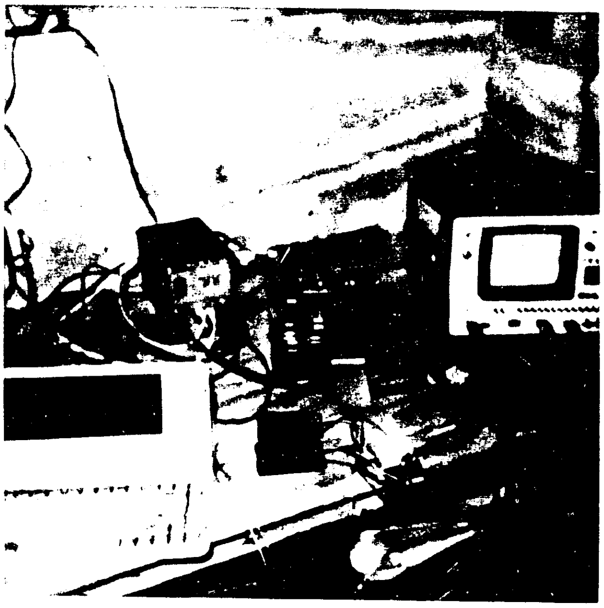
The program mentioned above can be written in a variety of languages assuming that the proper compilers are used. The Well-Head Controller program is written in BASIC language and compiled to Assembly language using BC705 version 2.11. The Assembly language is then compiled to machine language using a Motorola Freeware Assembler. The machine language code is then converted to the S-Record format and is ready to be loaded into the evaluation module. A copy of the Well-Head Controller BASIC program is included.

SCHEMATIC OF WELL-HEAD CONTROLLER





Motorola Evaluation Module



Microcomputer interface to Evaluation Module and Logic Analyzer for programming and target system debugging

Microprocessor Based Remote Oil Field Monitor and Controller(left)
Radio Frequency Data Link Terminal (right)

WELL-HEAD CONTROLLER PARTS LIST

RESISTORS:	
R1,R15:	100K,1/4W,5% CARBON FILM
R2,R4,R5,R7,R7,R8,R9, R11,R12,R19,R22:	10K,1/4W,5% CARBON FILM
R3:	10M,1/4W,5% CARBON FILM
R6:	1K,1/4W,5% CARBON FILM
R10:	18,1W,1% METAL FILM
R13:	5.1K,1/4W,5% CARBON FILM
R14:	510K,1/4W,5% CARBON FILM
R16:	5.6K,1/4W,5% CARBON FILM
R17,R20:	470,1/4W,5% CARBON FILM
R18:	200K,1/4W,5% CARBON FILM
R21:	18K,1/4W,5% CARBON FILM
TM1:	350 OHM, DIGI-KEY #PNT-110
CAPACITORS:	
C1:	100MF,15V, ALUMINUM ELECTROLITIC
C2,C3,C4,C5,C7,C12,C14, C19,C20:	.1MF,15V,CERAMIC DISC
C6,C16:	.01MF,15V,CERAMIC DISC
C8,C9:	22PF,15V,CERAMIC DISC
C10:	.02MF,15V,POLYSTYRENE
C11:	.005MF,15V,POLYSTYRENE
C13:	.0047MF,15V,POLYSTYRENE
C15,C18:	.1MF,15V,POLYSTYRENE
C17:	.022MF,15V,POLYSTYRENE
DIODES:	
D1,D4:	1N914
D2:	10V ZENER, 100mW
D3:	LM336Z-2.5,2.5 V ZENER
TRANSISTORS:	
Q1,Q2,Q3,Q5,Q7:	2N3904,NPN
Q4,Q6:	2N4036,PNP
VOLTAGE REGULATORS:	
VR1:	LM7805,5V,1 AMP
VR2:	LM7809,9V,1 AMP
INTEGRATED CIRCUITS:	
U1:	MC68HC705C8 MICROCONTROLLER
U2:	LM2904N DUAL LOW-POWER OP-AMP
U3:	UA555TC TIMER
U4:	XR-2211 PHASE LOCK LOOP FSK DEMODULATOR
U5:	CD4066B CMOS ANALOG MULTIPLEXER
U6:	MC7404 INVERTOR
BATTERY:	
B1:	12V,10 AMP-HR,RECHARGABLE

CRYSTAL:	
X1:	2 Mz
RELAY1:	120V,1 AMP CONTACTS
RC1:	12V COIL FOR RELAY1
TRANSCEIVER:	REALISTIC TRC-503, CAT# 21-402

```
'WELL-HEAD CONTROLLER PROGRAM
'Date: 01-03-92
'This program does the following:
'     - Controls mode timing and sequencing for a pumping system.
'     - Provides remote host computer or remote user the ability to
'     setup and control the Well-Head Controller.
'     - Performs necessary transceiver control tasks for radio
'     frequency digital transmissions.
'     - Allows for single radio frequency mulit-nodal network
'     transmissions.
'
START CONSTANT
START VARIABLE
    DECLARE VARIABLE x,y,i,j,k,it,jt,kt,h,m,s,hd,ld
    DECLARE VARIABLE HSLEEP,MSLEEP,SSLEEP
    DECLARE VARIABLE HSAMPLE,MSAMPLE,SSAMPLE
    DECLARE VARIABLE HPUMP,MPUMP,SPUMP
    DECLARE VARIABLE HEXHAUST,MEXHAUST,SEXHAUST
    DECLARE VARIABLE STATUS,ERROR,IX,IL,IH,HNODE,LNODE,NODE
    DECLARE VARIABLE NUMPUMP,NUMSAMP,NUMTERM,NUMINQ,REPORT
    DIM VARIABLE COMMAND(15)
START PROGRAM
    DECLARE STRING SLEEPS "Sleeping"
    DECLARE STRING SAMPLES "Sampling"
    DECLARE STRING PUMPS "Pumping"
    DECLARE STRING EXHAUSTS "Exhausting"
    DECLARE STRING SLEEPT "Sleep time   = "
    DECLARE STRING SAMPLET "Sample time  = "
    DECLARE STRING PUMPT "Pump time    = "
    DECLARE STRING EXHAUSTT "Exhaust time = "
    DECLARE STRING NUMPUMPT "Number of pumps ........... = "
    DECLARE STRING NUMSAMPT "Number of samples ......... = "
    DECLARE STRING NUMTERMT "Number of pump terminations = "
    DECLARE STRING NUMINQT  "Number of inquiries........ = "
CLOCK 2
NODE = 1
REPORT = 1
ld = NODE
CALL CONVERTD
HNODE = hd
LNODE = ld
OPENCOM 52 0 44
'sets all of port A as outputs
SETPORT A %11111111
'sets all of port B as inputs
SETPORT B %00000000
'turns all of port A off
OUTPORT A ALL 0
'sets initial mode timing values
HSLEEP = 0
MSLEEP = 0
SSLEEP = 5
HSAMPLE = 0
MSAMPLE = 0
```

```
SSAMPLE = 5
HPUMP = 0
MPUMP = 0
SPUMP = 5
HEXHAUST = 0
MEXHAUST = 0
SEXHAUST = 5
STATR:
NUMPUMP = 0
NUMSAMP = 0
NUMTERM = 0
NUMINQ = 0
COMMAND(0) = 0
OUTPORT A ALL 0
TOP:
RESETSP
'processes instructions
IF COMMAND(0) = 77
    x = COMMAND(2)
    IF x = 83
        GOTO SLEEP
    ENDIF
    IF x = 84
        GOTO SAMPLE
    ENDIF
    IF x = 80
        GOTO PUMP
    ENDIF
    IF x = 69
        GOTO EXHAUST
    ENDIF
    IF x = 82
        GOTO STATR
    ENDIF
ENDIF
GOTO SLEEP

SUB ACTCOM
    ERROR = 0
    x = COMMAND(0)
    IF x = 83
        CALL DISECT
        HSLEEP = h
        MSLEEP = m
        SSLEEP = s
        GOTO ENDIT
    ENDIF
    IF x = 84
        CALL DISECT
        HSAMPLE = h
        MSAMPLE = m
        SSAMPLE = s
        GOTO ENDIT
    ENDIF
```

```
    IF x = 80
        CALL DISECT
        HPUMP = h
        MPUMP = m
        SPUMP = s
        GOTO ENDIT
    ENDIF
    IF x = 69
        CALL DISECT
        HEXHAUST = h
        MEXHAUST = m
        SEXHAUST = s
        GOTO ENDIT
    ENDIF
    IF x = 63
        CALL DUMP
        GOTO ENDIT
    ENDIF
    IF x = 76
        REPORT = 1
        GOTO ENDIT
    ENDIF
    IF x = 79
        REPORT = 0
        GOTO ENDIT
    ENDIF
    IF x = 13
        GOTO ENDIT
    ENDIF
    ERROR = 1
ENDIT:
ENDSUB

SUB DUMP
'    Prints    sleep,sample,pump,exhaust    time
settings,numpump,numsamp,numterm.numinq
' Turns transmitter ON
    OUTPORT A 2 1
    FOR i = 1 TO 10
        DELAYM 100
    NEXT i
    OUTCOM 13
    OUTCOM 10
    OUTCOM 13
    OUTCOM 10
    FOR i = 1 TO 15
        x = i - 1
        OUTCOM SLEEPT(x)
    NEXT i
    h = HSLEEP
    m = MSLEEP
    s = SSLEEP
    CALL TPRINT
    OUTCOM 13
```

```
OUTCOM 10
FOR i = 1 TO 15
     x = i - 1
     OUTCOM SAMPLET(x)
NEXT i
h = HSAMPLE
m = MSAMPLE
s = SSAMPLE
CALL TPRINT
OUTCOM 13
OUTCOM 10
FOR i = 1 TO 15
     x = i - 1
     OUTCOM PUMPT(x)
NEXT i
h = HPUMP
m = MPUMP
s = SPUMP
CALL TPRINT
OUTCOM 13
OUTCOM 10
FOR i = 1 TO 15
     x = i - 1
     OUTCOM EXHAUSTT(x)
NEXT i
h = HEXHAUST
m = MEXHAUST
s = SEXHAUST
CALL TPRINT
OUTCOM 13
OUTCOM 10
FOR i = 1 TO 30
     x = i - 1
     OUTCOM NUMPUMPT(x)
NEXT i
ld = NUMPUMP
CALL CONVERTD
OUTCOM hd
OUTCOM ld
OUTCOM 13
OUTCOM 10
FOR i = 1 TO 30
     x = i - 1
     OUTCOM NUMSAMPT(x)
NEXT i
ld = NUMSAMP
CALL CONVERTD
OUTCOM hd
OUTCOM ld
OUTCOM 13
OUTCOM 10
FOR i = 1 TO 30
     x = i - 1
     OUTCOM NUMTERMT(x)
```

```
    NEXT i
    ld = NUMTERM
    CALL CONVERTD
    OUTCOM hd
    OUTCOM ld
    OUTCOM 13
    OUTCOM 10
    FOR i = 1 TO 30
        x = i - 1
        OUTCOM NUMINQT(x)
    NEXT i
    ld = NUMINQ
    CALL CONVERTD
    OUTCOM hd
    OUTCOM ld
    OUTCOM 13
    OUTCOM 10
'turns transmitter off
    OUTPORT A 2 0
ENDSUB

SUB TPRINT
'given h,m,s, prints hh:mm:ss
    ld = h
    CALL CONVERTD
    OUTCOM hd
    OUTCOM ld
    OUTCOM 58
    ld = m
    CALL CONVERTD
    OUTCOM hd
    OUTCOM ld
    OUTCOM 58
    ld = s
    CALL CONVERTD
    OUTCOM hd
    OUTCOM ld
ENDSUB


SUB CONVERTD
'given ld number converts to hd and ld ASCII equivalents
    x = ld
    y = ld
    hd = ld / 10
    hd = hd + 48
    x = x / 10
    x = x * 10
    x = y - x
    ld = x + 48
ENDSUB


SUB DISECT
```

```
'***** accepts high digit and low digit and returns numerical
'equivalent
'***** in ld
    hd = COMMAND(2)
    ld = COMMAND(3)
    CALL CONVERTN
    h = ld
    hd = COMMAND(5)
    ld = COMMAND(6)
    CALL CONVERTN
    m = ld
    hd = COMMAND(8)
    ld = COMMAND(9)
    CALL CONVERTN
    s = ld
ENDSUB

SUB CONVERTN
    hd = hd - 48
    hd = hd * 10
    ld = ld - 48
    ld = ld + hd
ENDSUB


SUB GETCOM
' note: disableint before getcom and enableint after
    OUTPORT A 2 0
    FOR i = 1 TO 10
        DELAYM 100
    NEXT i
    x = 0
    i = 0
    DO WHILE x <> 13
        IF i < 14
            x = INCOM
        ELSE
            x = 13
        ENDIF
        COMMAND(i) = x
        i = i + 1
    LOOP
' Turns transmitter OFF
    OUTPORT A 2 0
ENDSUB

SLEEP:
ENABLEINT
IF REPORT = 1
    OUTPORT A 2 1
    FOR i = 1 TO 10
        DELAYM 100
    NEXT i
    FOR i = 1 TO 8
```

```
            x = i - 1
            OUTCOM SLEEPS(x)
        NEXT i
        OUTCOM 13
        OUTCOM 10
        FOR i = 1 TO 10
            DELAYM 100
        NEXT i
        OUTPORT A 2 0
    ENDIF
    STATUS = 0
    OUTPORT A ALL 0
    OUTPORT A 4 1
    h = HSLEEP
    m = MSLEEP
    s = SSLEEP
    CALL TIME
    OUTPORT A 4 0
    GOTO SAMPLE

    SAMPLE:
    IF NUMSAMP < 99
        NUMSAMP = NUMSAMP + 1
    ENDIF
    ENABLEINT
    IF REPORT = 1
        OUTPORT A 2 1
        FOR i = 1 TO 10
            DELAYM 100
        NEXT i
        FOR i = 1 TO 8
            x = i - 1
            OUTCOM SAMPLES(x)
        NEXT i
        OUTCOM 13
        OUTCOM 10
        FOR i = 1 TO 10
            DELAYM 100
        NEXT i
        OUTPORT A 2 0
    ENDIF
    STATUS = 1
    OUTPORT A 1 1
    OUTPORT A 5 1
    h = HSAMPLE
    m = MSAMPLE
    s = SSAMPLE
    CALL TIME
    x = INPORT b 0
    IF x = 1
        OUTPORT A 5 0
        GOTO PUMP
    ELSE
        OUTPORT A 5 0
```

```
    GOTO SLEEP
ENDIF


PUMP:
IF NUMPUMP < 99
    NUMPUMP = NUMPUMP + 1
ENDIF
ENABLEINT
IF REPORT = 1
    OUTPORT A 2 1
    FOR i = 1 TO 10
        DELAYM 100
    NEXT i
    FOR i = 1 TO 7
        x = i - 1
        OUTCOM PUMPS(x)
    NEXT i
    OUTCOM 13
    OUTCOM 10
    FOR i = 1 TO 10
        DELAYM 100
    NEXT i
    OUTPORT A 2 0
ENDIF
STATUS = 2
OUTPORT A 1 0
OUTPORT A 0 1
OUTPORT A 6 1
h = HPUMP
m = MPUMP
s = SPUMP
CALL TIME
OUTPORT A 0 0
OUTPORT A 6 0
GOTO EXHAUST

EXHAUST:
ENABLEINT
IF REPORT = 1
    OUTPORT A 2 1
    FOR i = 1 TO 10
        DELAYM 100
    NEXT i
    FOR i = 1 TO 10
        x = i - 1
        OUTCOM EXHAUSTS(x)
    NEXT i
    OUTCOM 13
    OUTCOM 10
    FOR i = 1 TO 10
        DELAYM 100
    NEXT i
    OUTPORT A 2 0
```

```
ENDIF
STATUS = 3
OUTPORT A 1 0
OUTPORT A 0 0
OUTPORT A 7 1
h = HEXHAUST
m = MEXHAUST
s = SEXHAUST
CALL TIME
OUTPORT A 7 0
GOTO SLEEP

'timing delay subroutine
SUB TIME
    x = 0
    DO WHILE h > 0
        FOR kt = 1 TO 60
            CALL DMINUTE
'********* allows for termination of pumping cycle ****
            IF x = 1
                GOTO EXIT
            ENDIF
'******************************************************
        NEXT kt
        h = h - 1
    LOOP

    DO WHILE m > 0
        CALL DMINUTE
        m = m - 1
    LOOP

    DO WHILE s > 0

'******* allows for termination of pump cycle *******
        IF STATUS = 2
            x = INPORT b 1
            IF x = 1
                IF NUMTERM < 99
                    NUMTERM = NUMTERM + 1
                ENDIF
                GOTO EXIT
            ENDIF
        ENDIF
'****************************************************

        FOR it = 1 TO 10
            DELAYM 100
        NEXT it
        s = s - 1
    LOOP
EXIT:
ENDSUB
```

```
SUB DMINUTE
    FOR jt = 1 TO 60

'****** allows for termination of pumping cycle *******
        IF STATUS = 2
            x = INPORT b 1
            IF x = 1
                IF NUMTERM < 99
                    NUMTERM = NUMTERM + 1
                ENDIF
                GOTO OVER
            ENDIF
        ENDIF
'*************************************************

        FOR it = 1 TO 10
            DELAYM 100
        NEXT it
    NEXT jt
OVER:
ENDSUB
'processes instructions from SCI port(r.f. transmissions)
INT SCI
    DISABLEINT
    OUTPORT A 2 0
    IX = INCOM
    IF IX = 27
        IH = INCOM
        IL = INCOM
        IF IH = HNODE
            IF IL = LNODE
                IF NUMINQ < 99
                    NUMINQ = NUMINQ + 1
                ENDIF
                CALL GETCOM
                CALL ACTCOM
                ENABLEINT
                GOTO TOP
            ENDIF
        ENDIF
    ENDIF
ENDINT
```

SOLAR POWERED AIR PULSE OIL PUMP

A block diagram of a solar powered Air Pulse Oil Pump is contained in Figure 1. It consists of a solar collector connected to a twelve volt compressor, similar to the compressors used to plug into automobile cigarette lighter socket to pump up the air in a tire except this compressor is industrial grade. A regulator stores excess electricity in a deep discharge twelve volt battery which has a capacity of about 75 ampere-hour.

The energy to lift a barrel of water a height of 150 meters, assuming 100 % efficiency is about 233,000 joules. Using a solar constant of 1.92 calories per square centimeter per minute and a one square meter collector, a 10 % efficiency for converting the light to electricity yields as follows:

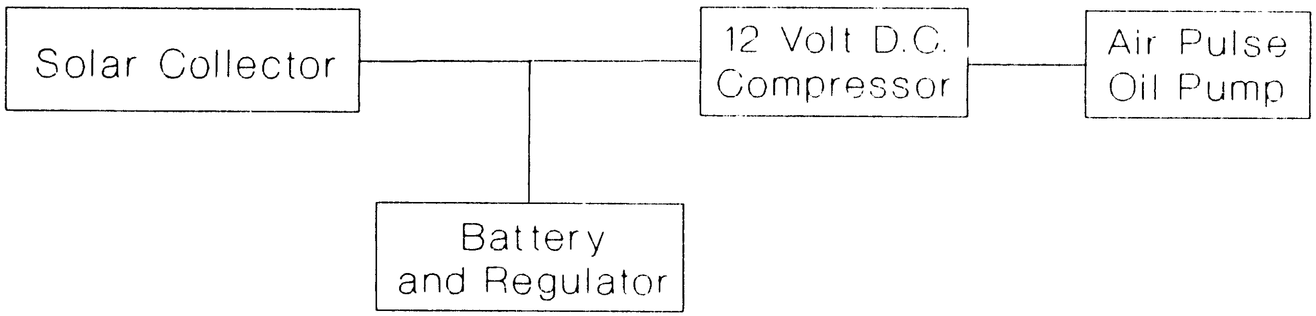
$$\text{Energy} = (1.92\ \text{cal-cm}^{-2}\text{-min}^{-1})(4.18\text{J/cal})(1\ \text{m}^2)(10^4\ \text{cm}^2/\text{m}^2) \times$$

$$(8\ \text{hrs})(60\ \text{min/hr})(\ 0.1\ \text{eff.}) = 3{,}852{,}288\ \text{Joules.}$$

This rough calculation indicates that the system should be cap0able of pumping at least one barrel per day from shallow wells to a depth of 500 feet.
The energy stored in a 75 ampere-hour battery should be about 3,000,000 joules of energy when it is fully charged. This should be enough reserve to pump at least one barrel of liquid to the surface.

In the event that the sun does not shine for several days, the system will not be able to pump during most of that period.

FIGURE 1.
BLOCK DIAGRAM OF A SOLAR POWERED AIR PULSE OIL PUMP



END

DATE
FILMED
3/3/92

I