Exploring 1-wire devices on the GA144

by

Franklin Amador

Outline

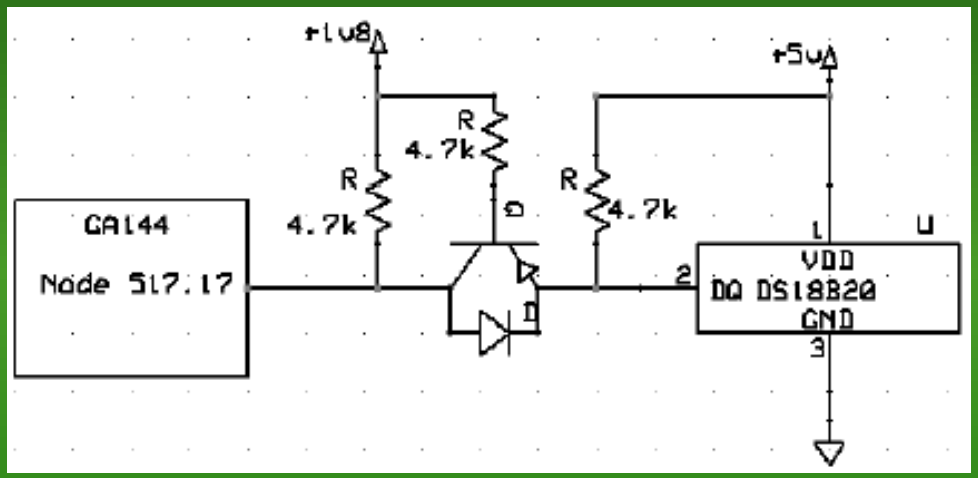
- Introduction
- The 1-wire Interface
- The 1-wire Protocol
- arrayForth® Implementation
- polyForth® Implementation
  - Snorkel & Ganglia Integration
- Results
- Conclusions

Introduction

- Why?
  - Port code from camel Forth
  - Expand I/O on Node 606 for chip select on SPI
  - Because I can
  - Low Speed Protocol
    - Standard : 16.3 kbps
    - Overdrive : 10 times Standard
- Results:
  - Inadvertently opened up 1-wire market for Green Arrays.

1-wire Interface Circuit

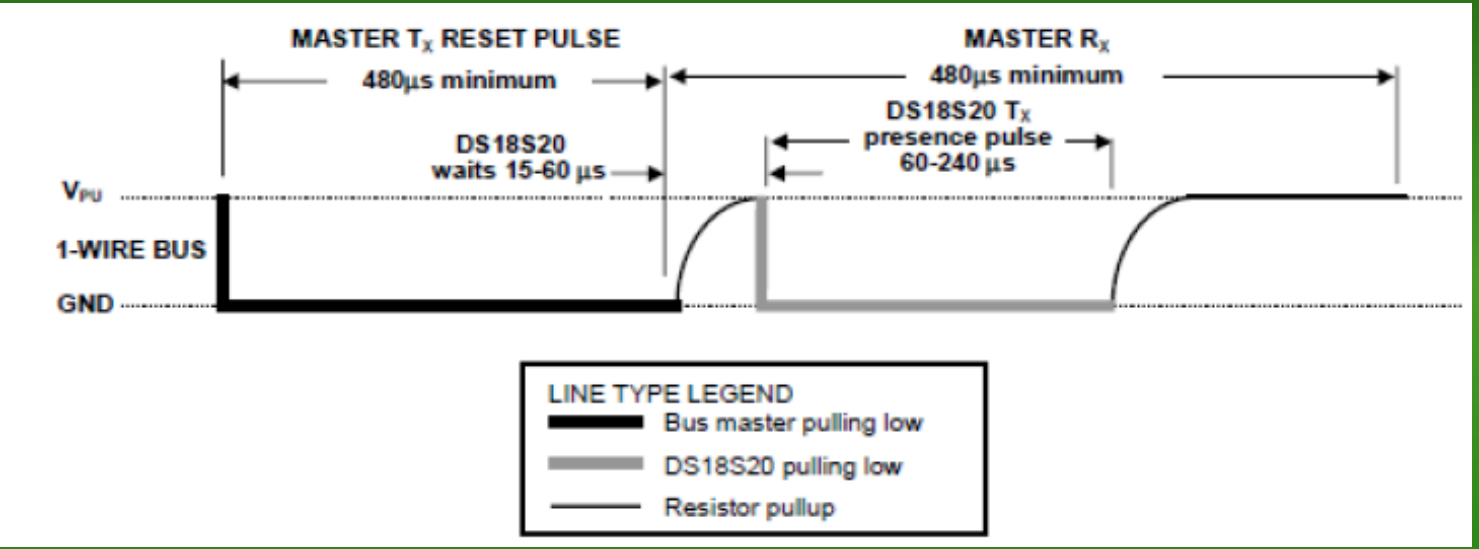
- Voltage Converter

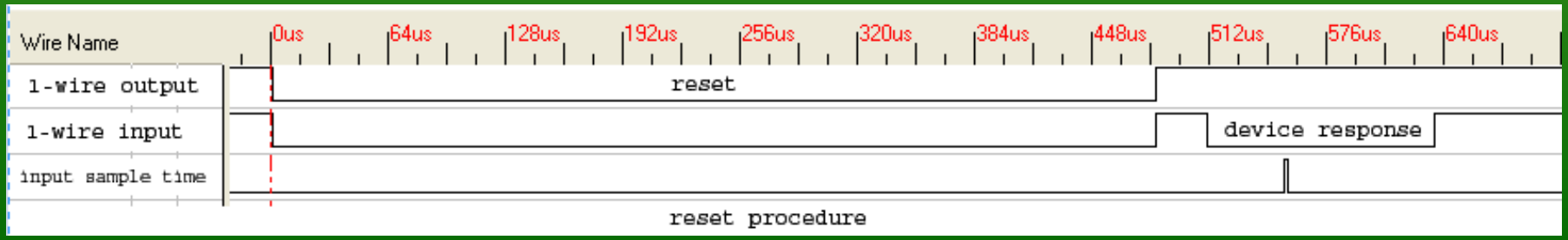


- Recommendation:
  - Use Evaluation Board TI's TXB0108 Level Shifter

1-wire Protocol

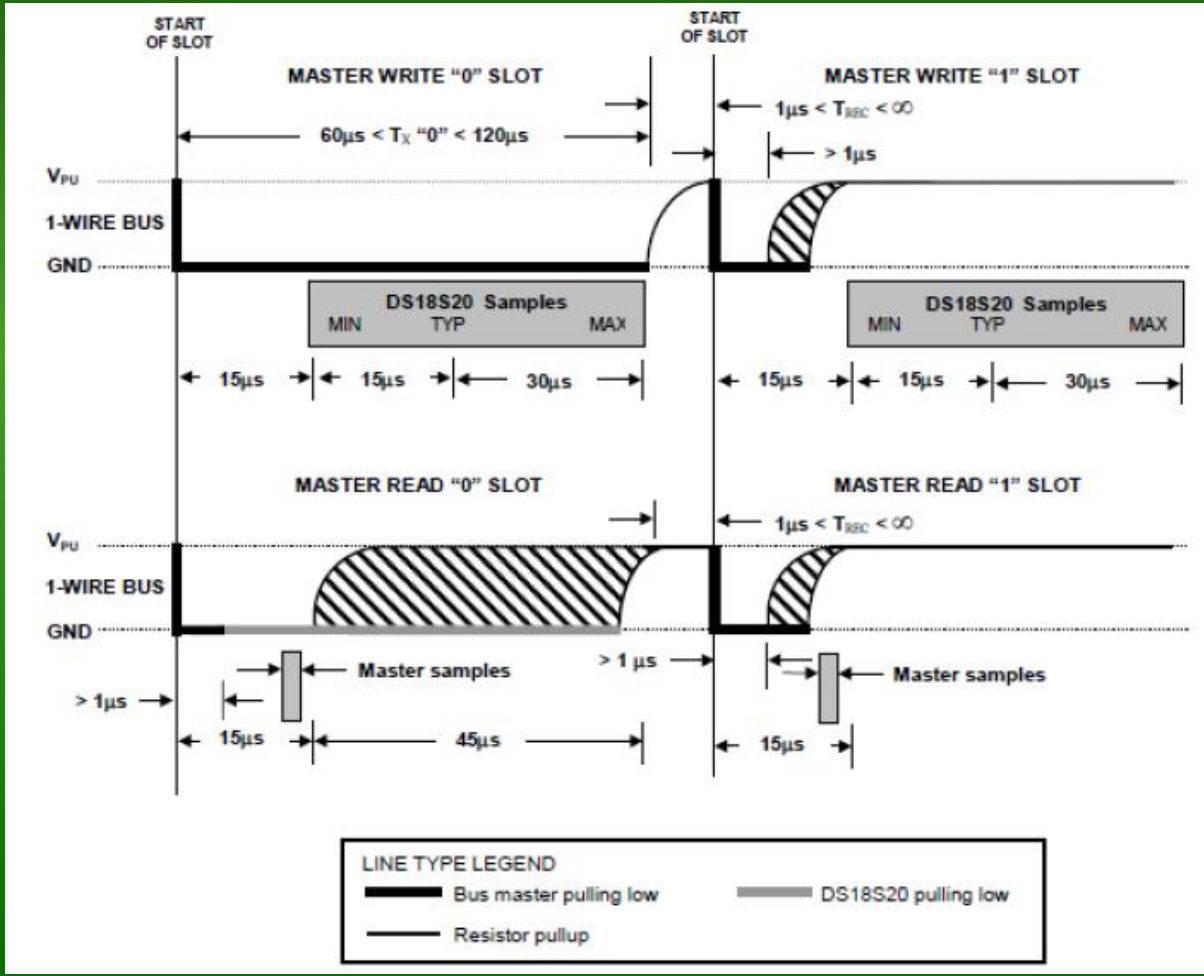
- Initialization Timer: Presence & Reset Pulse





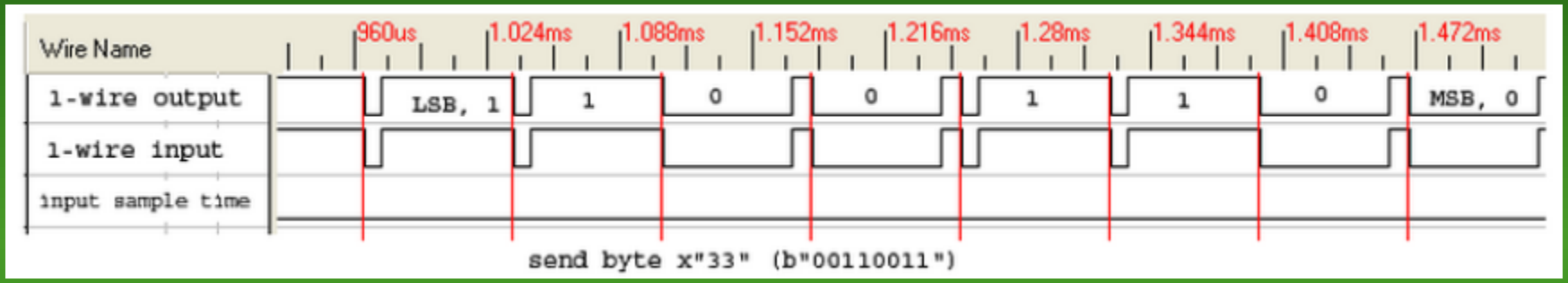
1-wire Protocol cont.

- Read/Write Time Slot Timing Diagram

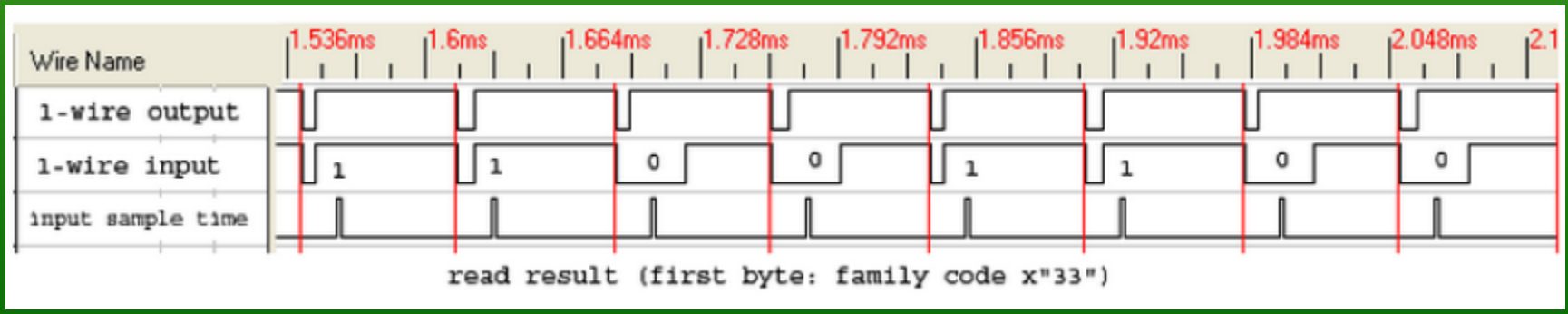


Sending & Receiving Timing

- Sending x33 example



- Receiving x33 example



arrayForth® Interface

- 1-wire Kernel

```
onewire dallas-1wire protocol,


dly timed usec loop delay
wat wait 500 usec
?pin read pin.17 value
hiz set pin.17 to high-impedance
low set pin.17 low
reset -n init device, presence pulse br

slot read/write 1bit to/from 1wire bus br

send sample pin.17, if x100 xor else br

receive right-shift main word, delay
pfreset send presence pulse back
pfslot send slot result back
pftouch touch read/write @/! from node 601
```

```
864 list
onewire -- kernel,

517 node 0 org
delay 00 for . . unext ;
wait 02 100000 delay ;
?pin 04 @b 20000 and ;
hiz 06 0 !b ;
low 08 20000 !b ;
reset 0a -n low wait hiz 15000 delay ?pin wait
.... 11 if dup or ; 13 then - ;
slot 14 n-n low 1200 delay dup 1 and .,
.... 19 if hiz 1b then,
send 1b drop 3800 delay ?pin,
.... 1f if over 100 or over . . . 23 then,
recv 23 drop 2/ 7000 delay hiz 400 delay ;
pfreset 29 -n reset ! ;
pfslot 2b n-n @ slot ! ;
pftouch 2d n-n @ 7 for slot next ! ; 32
```

polyForth® Interface

- polyForth® to GA144 Interface

```
301                    0 Refs    0 Other blocks
 0 ( Basic 1-wire operations )
 1 HEX
 2 ND N516 2 uu 8 rr 0 ,path
 3 1D5 CONSTANT RIGHT
 4 : OWRESET ( - n) N516 2029 1 RIGHT R!@ SWAP DROP 3 / ;
 5 : OWSLOT ( d - d) N516 202B 1 RIGHT R! 0 RIGHT R!@ DROP ;
 6 : OWTOUCH ( d - d) N516 202D 1 RIGHT R! 0 RIGHT R!@ DROP ;
 7 : OWPUT ( d-) OWTOUCH DROP ;
 8 : OWGET ( -n) FF OWTOUCH ;
 9
10 ( Use only if there is a single 1-wire device attached )
11 : SHOWID ( - ) OWRESET IF 033 OWPUT 7 FOR OWGET . NEXT THEN ;
12
13 ( use only if there are no single 1-wire devices attached )
14 : TEST ( - ) OWRESET . 0 OWSLOT DROP 55 OWPUT
15       7 FOR OWGET . NEXT ;
```

polyForth® Interface cont.

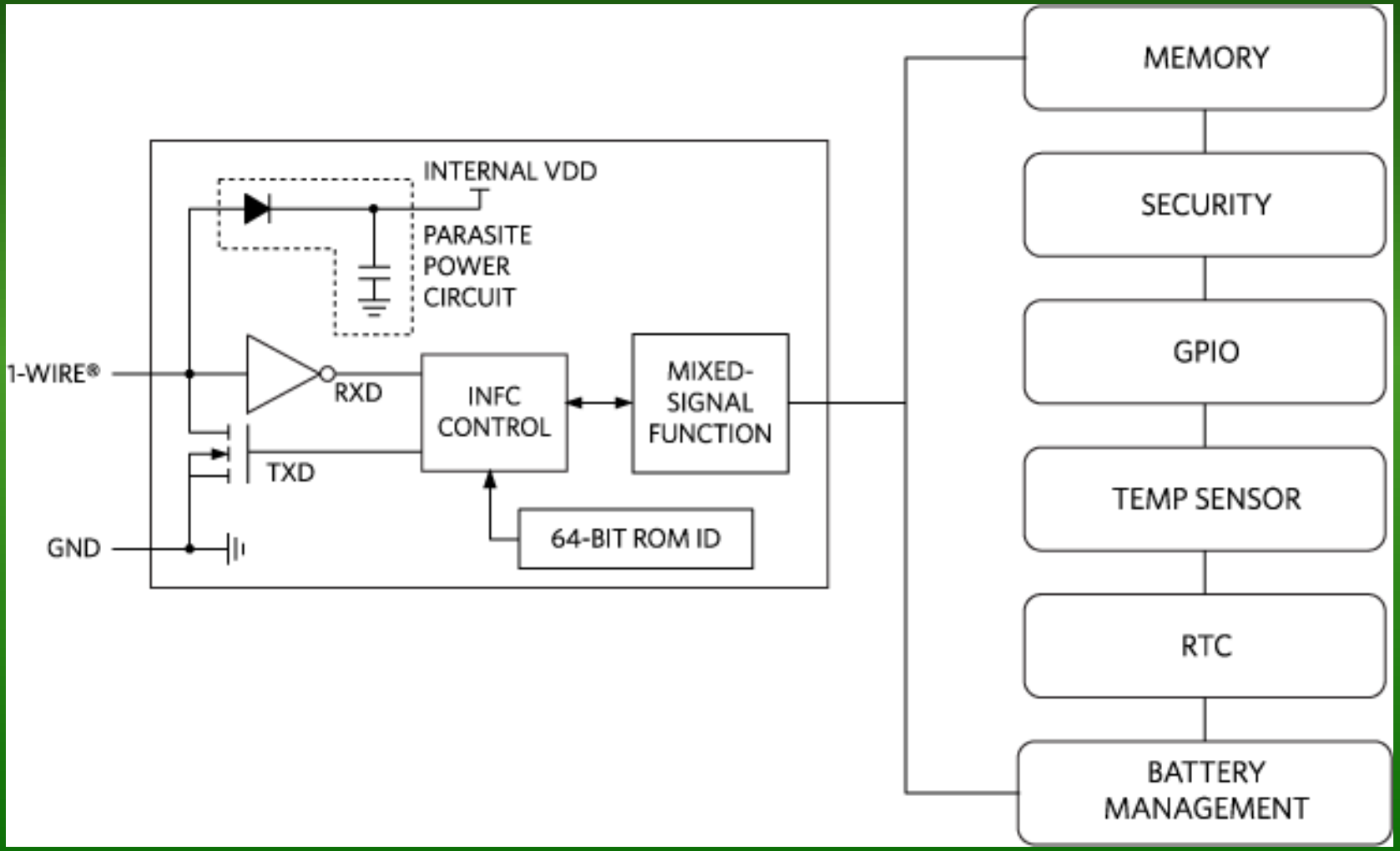
```
305                    0 Refs     0 Other blocks
 0 ( Maxim 1-wire high level operations )
 1 HEX
 2 : SHOWIDS ( -- ) NEWSEARCH
 3   BEGIN ROMSEARCH CR ROMID 8 + ROMID DO I C@ 3 U.R LOOP
 4   0= UNTIL CR ;
 5
 6 : SENDID ( addr -- )  OWRESET
 7    IF 55 OWPUT 8 OVER + SWAP DO I @ OWPUT LOOP
 8   ELSE ." failed" DROP THEN ;
 9
10 : READSCRATCH ( a - ) SENDID BE OWPUT 8 FOR OWGET . NEXT ;
11
12 : OWCONVERT ( a -- ) SENDID 44 OWPUT ;
13
14 : READTEMP ( a -- n ) SENDID BE OWPUT OWGET OWGET 8 LSHIFT OR ;
15
```

```
306                    0 Refs     0 Other blocks
 0 ( Maxim 1-wire high level operations )
 1 HEX
 2 : TEMP>PAD ( n -- ) 5 * DECIMAL DUP ABS 0
 3   <# # 2E HOLD # #S SIGN #> TYPE SPACE ;
 4
 5 : OWID ( --addr ) 8 OVER + SWAP DO I @ . LOOP ;
 6
 7 HEX CREATE SENSOR1 10 , FB , 24 , B8 , 2 , 8 , 0 , 22 ,
 8 HEX CREATE SENSOR2 10 , CA , B7 , B6 , 2 , 8 , 0 , 14 ,
 9
10 : FINAL1 SENSOR1 OWCONVERT 750 MS SENSOR1 READTEMP TEMP>PAD ;
11
12 : FINAL2 SENSOR2 OWCONVERT 750 MS SENSOR2 READTEMP TEMP>PAD ;
13
14 : EXAMPLE1 SENSOR1 OWCONVERT 750 MS SENSOR1 READSCRATCH ;
15
```

Results

```
SHOWIDS
 10 FB 24 B8  2  8  0 22
ok
SHOWIDS
 10 CA B7 B6  2  8  0 14
 10 FB 24 B8  2  8  0 22
ok
FINAL1 21.0 ok
FINAL2 20.5 ok
 ok
EXAMPLE1 42 0 75 70 255 255 15 16 64 ok
HEX EXAMPLE1 2A 0 4B 46 FF FF F 10 40 ok
_
```

Applications



Conclusion

- GA144 Architecture is Simple and Modular
- Porting Code was straight forward and simple
- Recommendations:
  - VHDL/Verilog to arrayForth® compiler and treat the F18 computer as a versatile logic unit.
  - Either Port arrayForth® into polyForth® or vice-versa for a simple one design software interface.

Questions?

- Contact:

  Franklin Amador

  fdamador@comcast.net

- Reference:
  - https://github.com/fdamador/GA144-1-Wire-Application-Note
  - http://en.wikipedia.org/wiki/1-Wire
  - http://www.maximintegrated.com