Q1 and other weird floppyformats

- Why FloppyTools exist

Poul-Henning Kamp

phk@FreeBSD.org

phk@Varnish.org

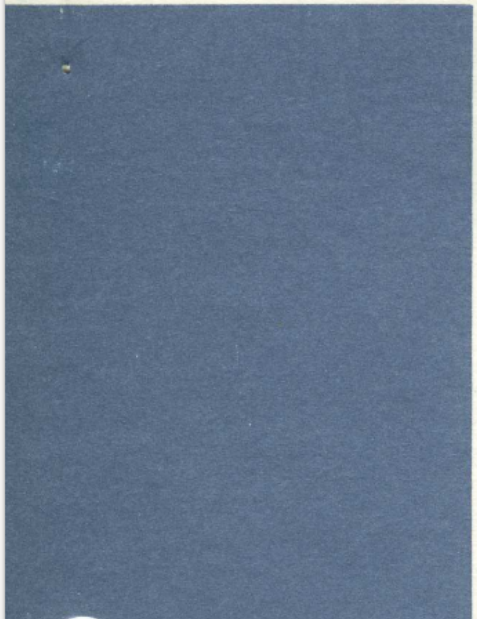
@bsdphk

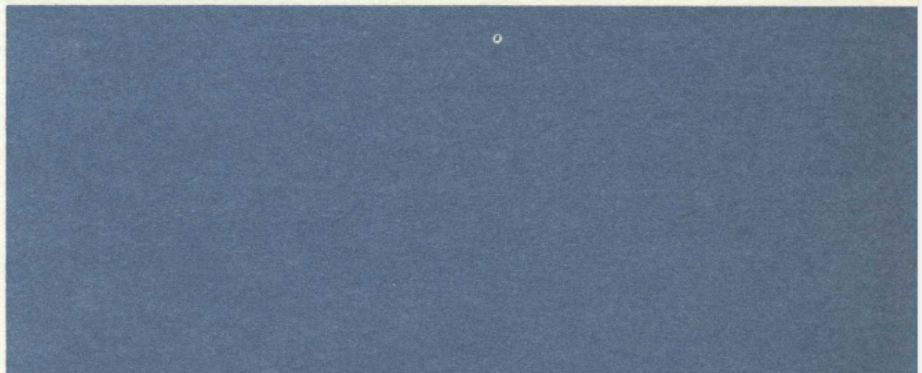
GA21-9182 aka "IBM 3740 format"

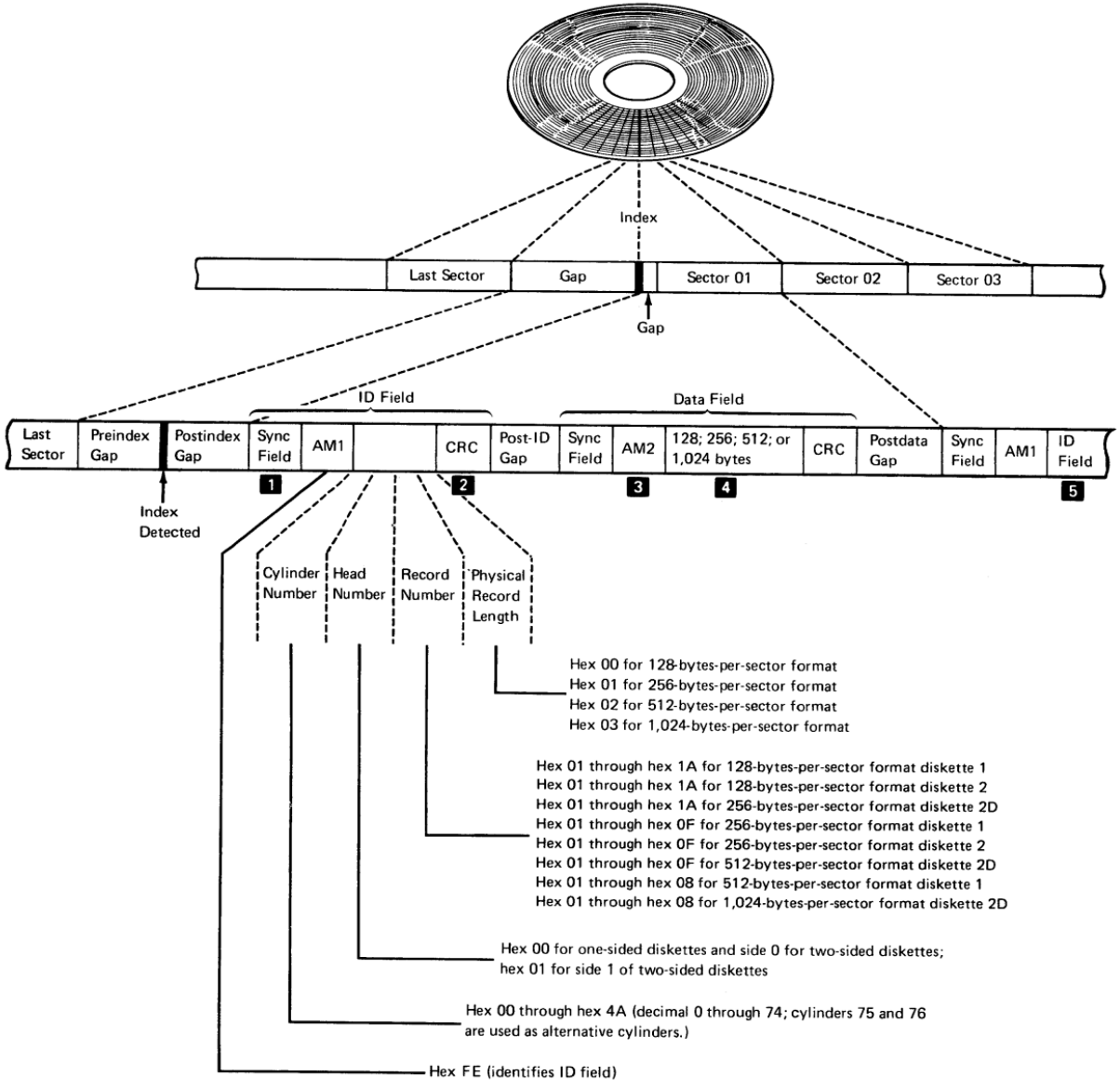
Product Reference Literature

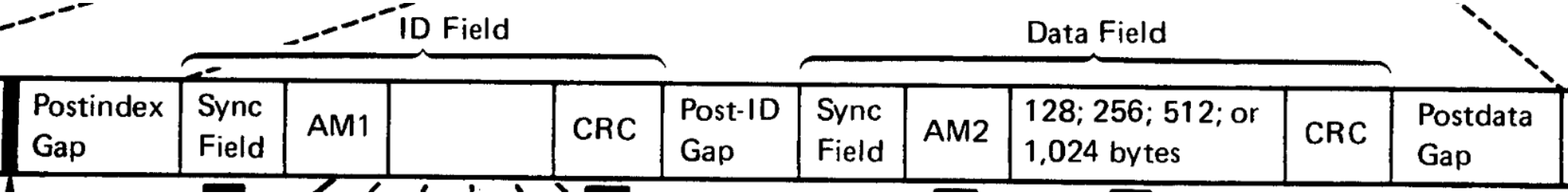
IBM

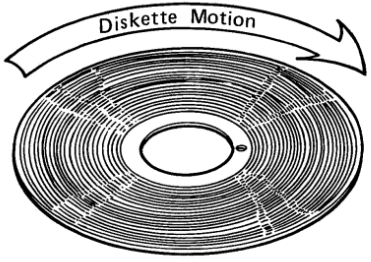
The IBM Diskette
General Information Manual

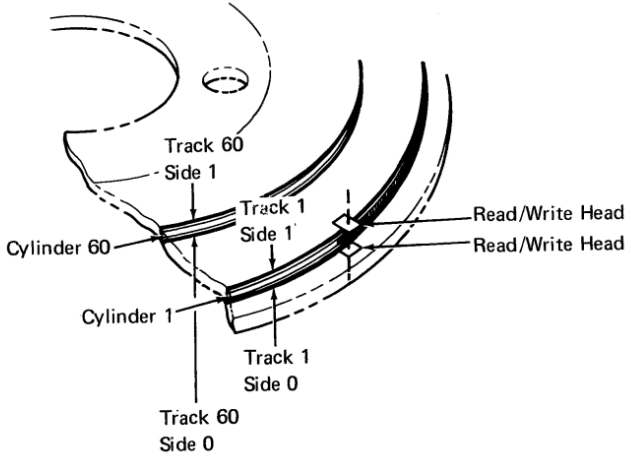





Index
Last Sector
Gap
Sector 01
Sector 02
Sector 03
Gap
ID Field
Data Field
Last Sector
Preindex Gap
Postindex Gap
Sync Field
AM1
CRC
Post-ID Gap
Sync Field
AM2
128; 256; 512; or 1,024 bytes
CRC
Postdata Gap
Sync Field
AM1
ID Field
1
2
3
4
5
Index Detected
Cylinder Number
Head Number
Record Number
Physical Record Length
Hex 00 for 128-bytes-per-sector format
Hex 01 for 256-bytes-per-sector format
Hex 02 for 512-bytes-per-sector format
Hex 03 for 1,024-bytes-per-sector format
Hex 01 through hex 1A for 128-bytes-per-sector format diskette 1
Hex 01 through hex 1A for 128-bytes-per-sector format diskette 2
Hex 01 through hex 1A for 256-bytes-per-sector format diskette 2D
Hex 01 through hex 0F for 256-bytes-per-sector format diskette 1
Hex 01 through hex 0F for 256-bytes-per-sector format diskette 2
Hex 01 through hex 0F for 512-bytes-per-sector format diskette 2D
Hex 01 through hex 08 for 512-bytes-per-sector format diskette 1
Hex 01 through hex 08 for 1,024-bytes-per-sector format diskette 2D
Hex 00 for one-sided diskettes and side 0 for two-sided diskettes; hex 01 for side 1 of two-sided diskettes
Hex 00 through hex 4A (decimal 0 through 74; cylinders 75 and 76 are used as alternative cylinders.)
Hex FE (identifies ID field)







Where to read & write

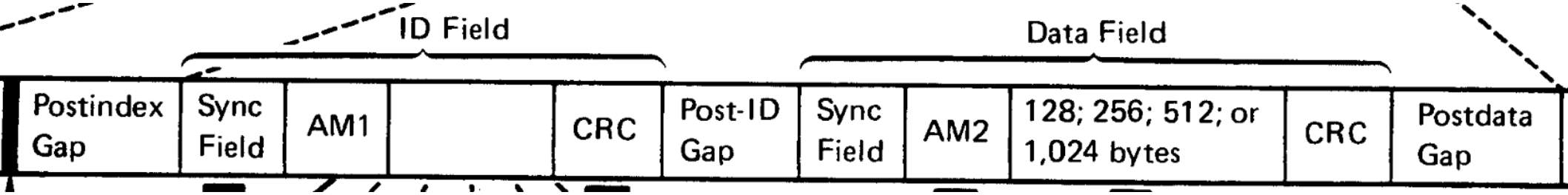
Step motor positions head on track.

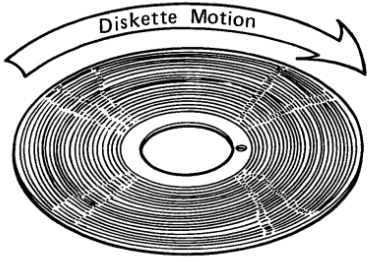
But not guaranteed.

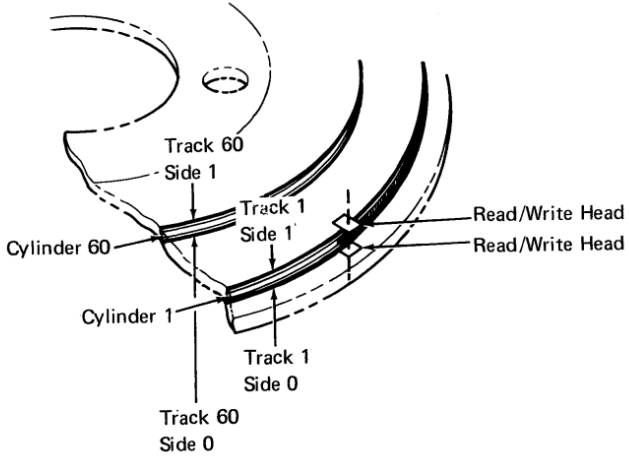
Electronics select top/bottom head.

This part just works.

Now we just need to sort out when







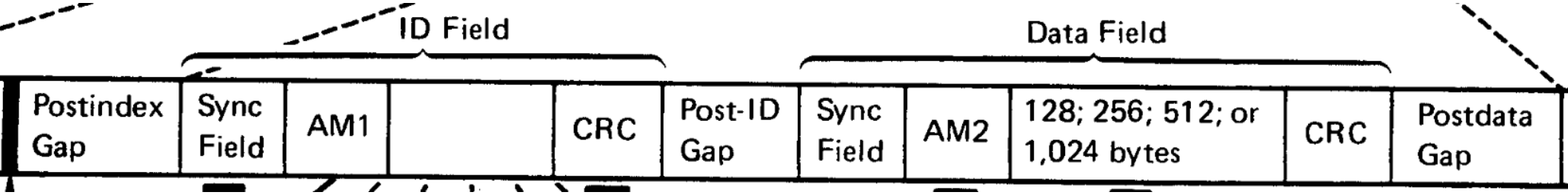
Index signal = 1/rev

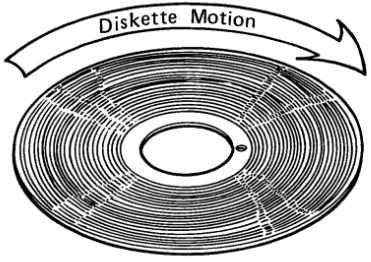
Hole in disc + photo-sensor

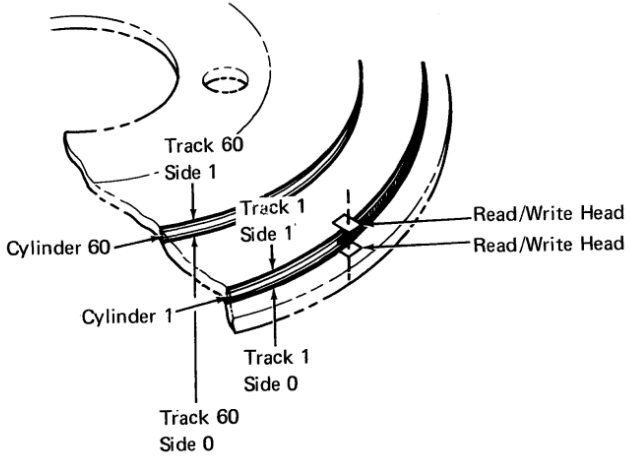
Rotation rate varies:

- Motor tolerance
- Grid frequency & voltage
- Disc-sleeve friction
- Disc-head friction

Solution: Insert generous ”GAP” fields





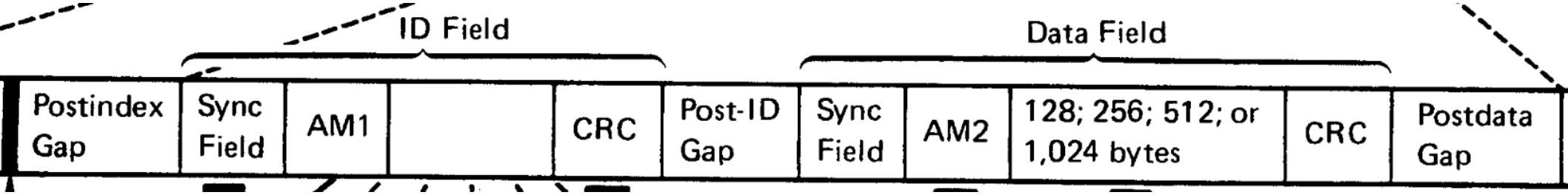


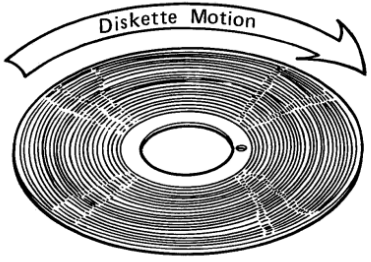
Rotation rate variation ⇒ data rate variation.

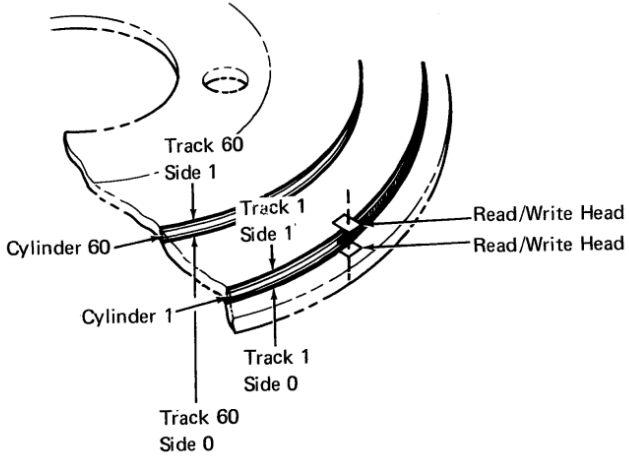
Sectors in track may be written on many systems, each with unique data-rate.

Solution:

Insert (generous) SYNC fields where data rate can be measured by reader.







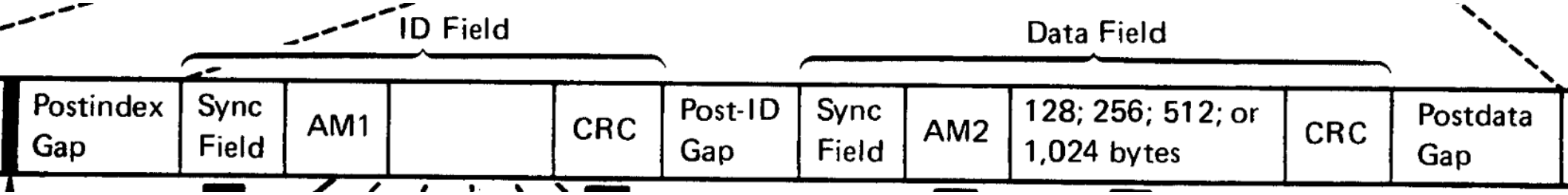
Stepper positioning fails - rarely, but it does happen.

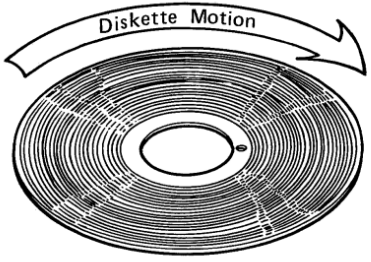
One sector per track is useless.

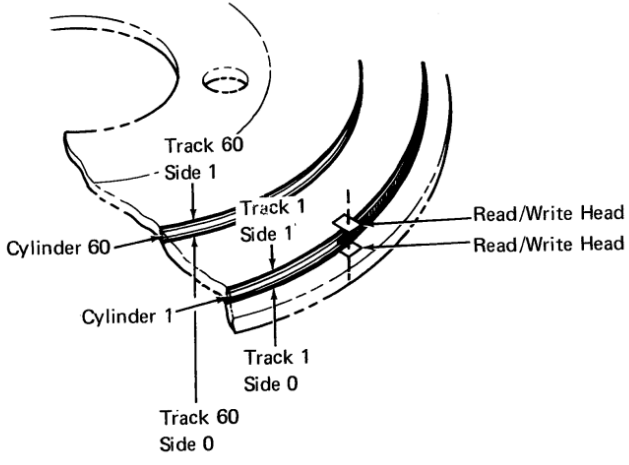
Solution: Address Marks

Splits track into sectors, and also contains track# and head#.

AM's written only during formatting







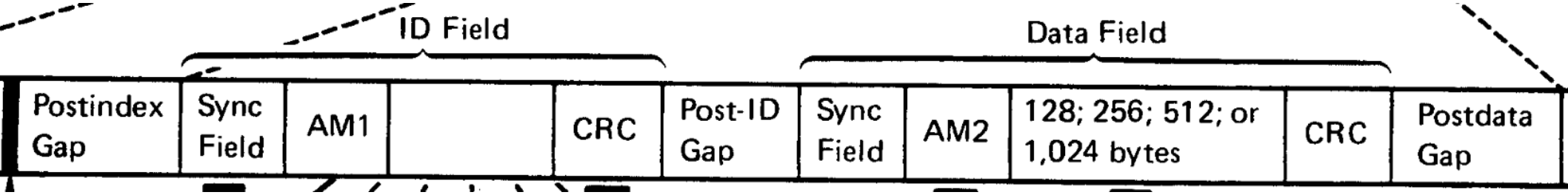
Things dont always work

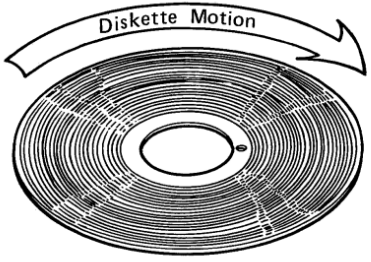
Head position, media imperfections, dirt, vibration, murphys law.

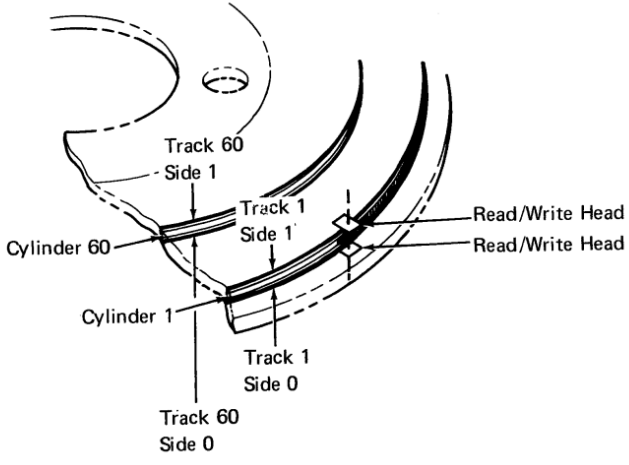
Solution:

Write checksum after information

(information = AM and sector content)







Writing a sector:

Move to cylinder & select head

Read AMs until we get the right one

Turn write current on

Write Sync+AM2+Data+CRC

Turn write current off

Add Post-ID/Data gaps to isolate write current transients.

Hard sectored disks - and why it is a semi-bad idea

Instead of all this complexity, punch a hole for each sector in the disk

Detect index pulse, count sector pulses, read or write, done.

Missing holes => read/write the wrong place.

- Some HS formats: write CHS in front of data

Still need gaps for speed variation, but fewer and smaller = higher density

- Very important for ”washing-machine” hard-disks
- ”Electronic holes” instead of physical holes for precision&reliability
- May or may not also have address marks

So why did some mfg. not use IBM's format ?

A) We want to be incompatible so we can sell preformatted media
   All of them I suspect ?

B) We can do better (= stuff more data onto a floppy)
   Intel ISIS, HP98xx, DEC RX02, Q1, Zilog, Commodore, Apple

C) The circuitry is far too expensive (no single-chip FDC yet)
   Data General

D) We do not belive in separation of concerns
   Q1, Zilog

WD introduces FD1771 floppy controller chip => New designs use IBM format.
   … except Apple and Commodore, and for backwards compat.

Not always smart to roll your own

Data general uses trivial checksum.

Competitors sales-people: ”Ridiculous”, ”Amateurs” &c

DG Implements their own CRC16 (cheaply!)

The worst possible CRC16, even worse than the trivial checksum:

16 Bit CRCs from https://users.ece.cmu.edu/~koopman/crc/crc16.html:

```
(0x8d95; 0x11b2b) <=> (0xd4d8; 0x1a9b1) {65519,1149,62,19,9,5,5} CRC-16F/3 ("215453p")
[…]
(0xd175; 0x1a2eb) <=> (0xd745; 0x1ae8b) {32751,32751,93,93,11,11,2,2} CRC-16F/4.2 ("321353")
[…]
(0x8080; 0x10101) <=> (0x8080; 0x10101) {8} | Data General floppy drive CRC
```

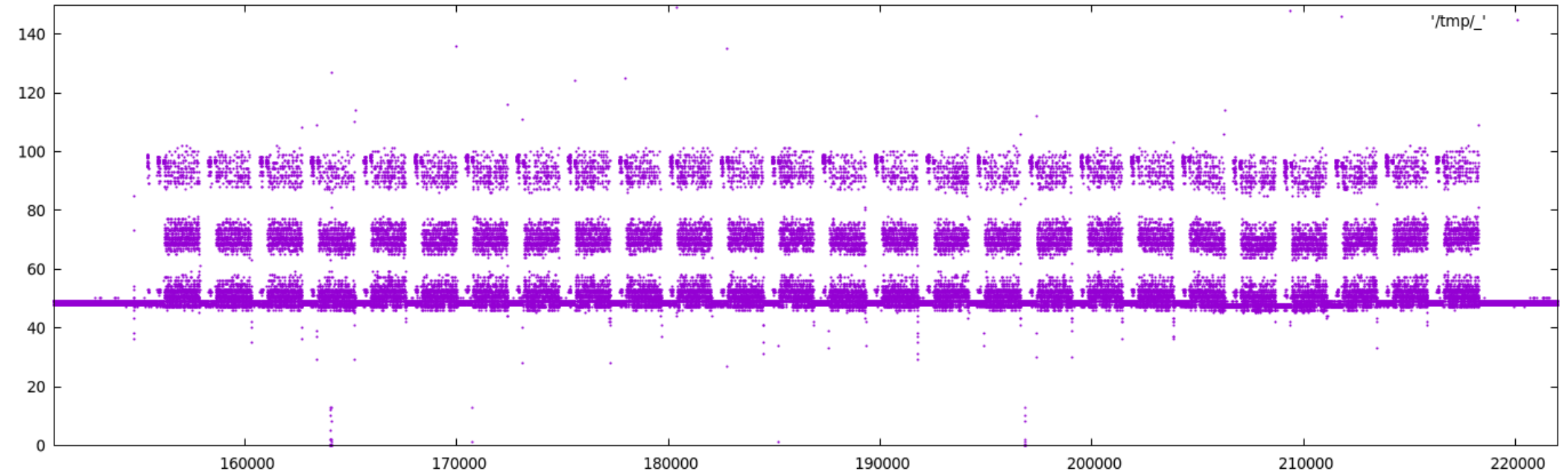
Flux-reading

When we read magnetic media we detect flux reversals.

The sign of the magnetic field is lost, we only get a pulse when it changes.

”Flux readers” like KryoFlux, GreaseWeazle measure time between flux reversals

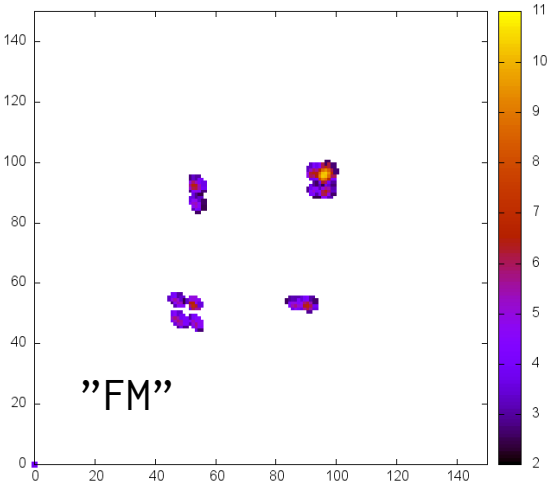
Your task, should you accept it, is to make sense of that...

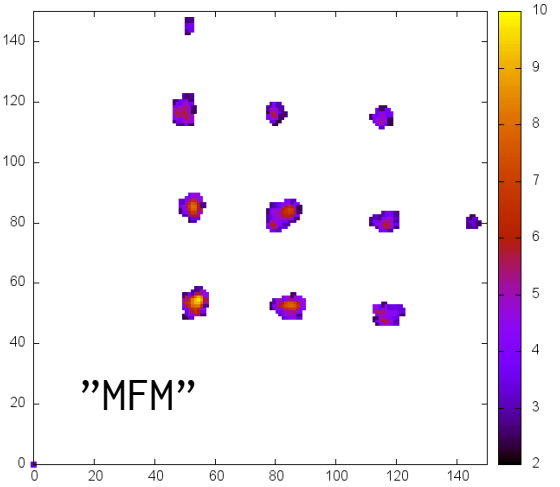


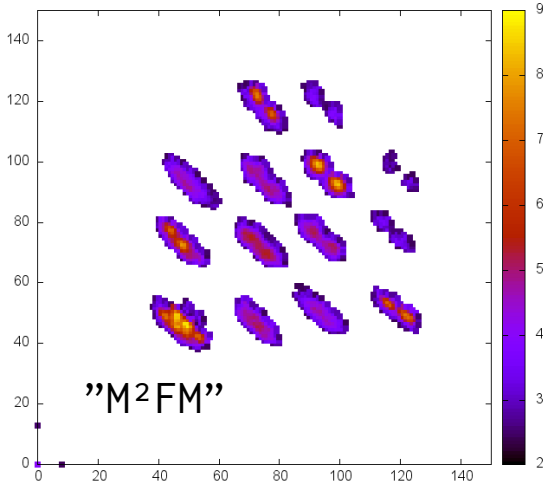
Modulations

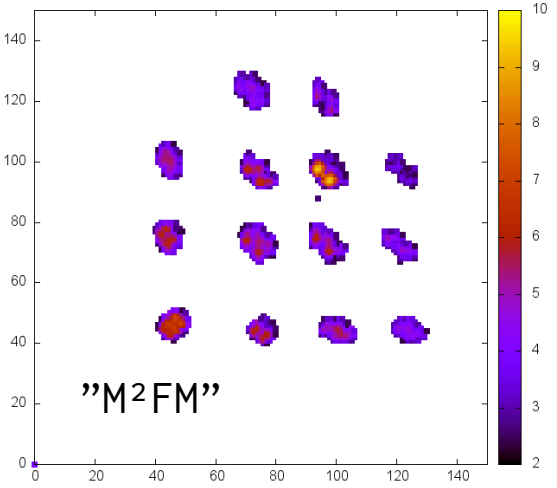
”Eye-Diagrams”

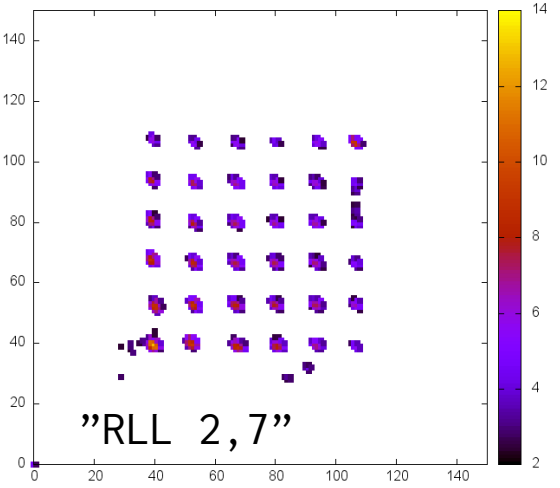
Plot dT[n] vs dT[n]











FloppyTools - principle of operation

Convert input data to convenient format

Split into AMs and sectors

Convert to bytes

Validate checksum

Keep track of what we have read, what’s missing, and conflicts

Simplest example: Zilog MCZ

```
crc_func = crcmod.predefined.mkCrcFun('crc-16-buypass')

class ZilogMCZ(media.Media):
    ''' ... '''

    SECTOR_SIZE = 136
    GEOMETRY = ((0, 0, 0), (77, 0, 31), SECTOR_SIZE)

    GAP = fluxstream.fm_gap(32)

    def process_stream(self, stream):
        schs = (stream.chs[0], stream.chs[1], 0)
        if not self.defined_chs(schs):
            return None

        flux = stream.fm_flux()
```

Simplest example: Zilog MCZ

```
        retval = False
        for data_pos in stream.iter_pattern(flux, pattern=self.GAP):
            data_pos -= 4

            data = stream.flux_data_fm(flux[data_pos:data_pos+((2+self.SECTOR_SIZE)*32)])
            if data is None:
                continue

            data_crc = crc_func(data)
            if data_crc != 0:
                continue

            chs = (data[1], 0, data[0] & 0x7f)
            if not self.defined_chs(chs):
                continue

            self.did_read_sector(chs, data[:-2], stream)
            retval = True
        return retval
```

Complex example: Q1

The basic format is pretty simple: AM + Sector + checksum (not CRC!)

But number & size of sectors are ”per track”.

If we can read track zero, we get catalog which tells us layout

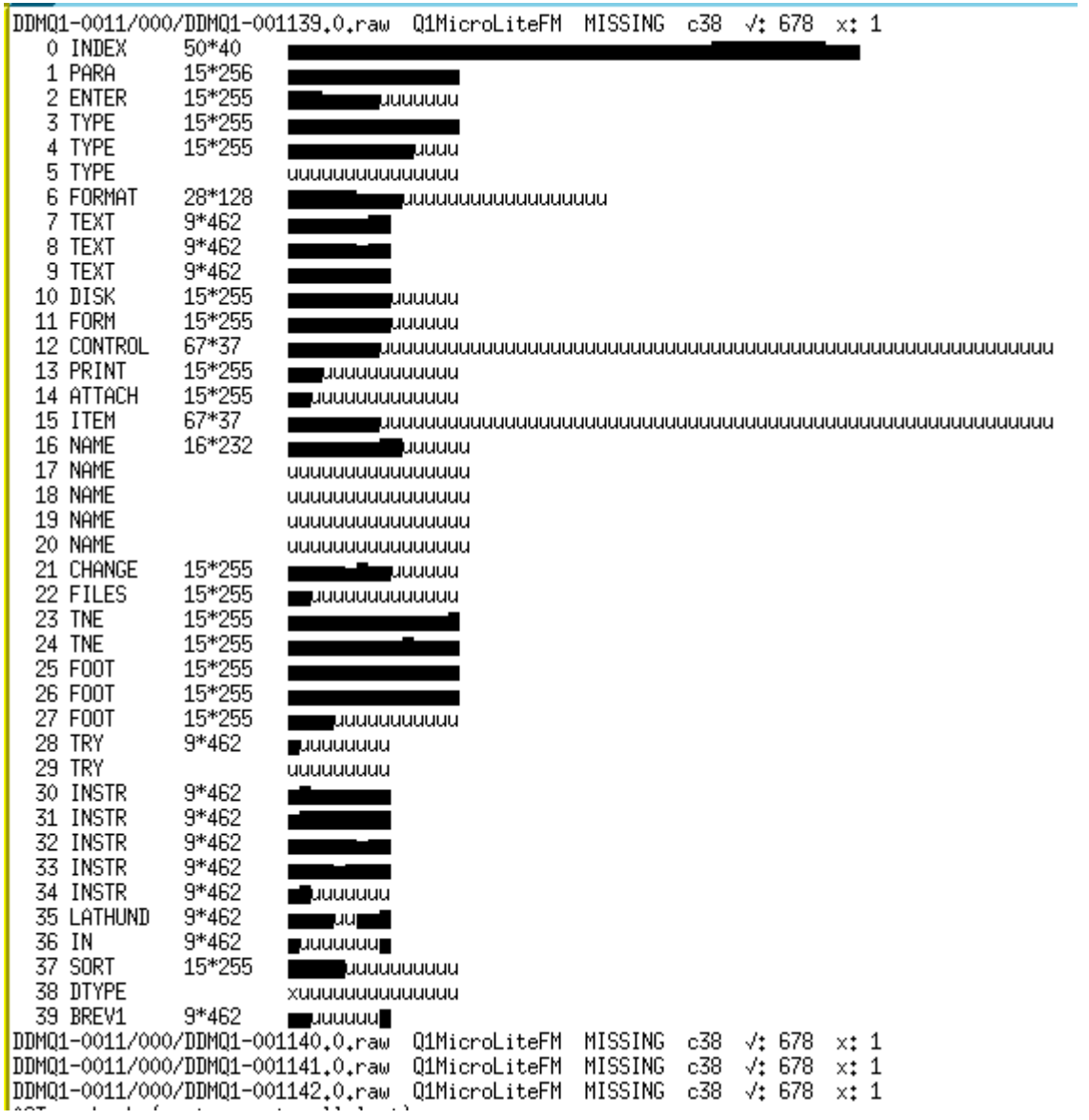
If we can not read track zero, we attempt heuristics.

Ohh, and one more thing…

Formatting does not write valid sectors, only the address marks are written

⇒ Bad checksums on unwritten (= unallocated?) sectors

But can be recognized because entire sector, and checksum is zero bits


DDMQ1-0011/000/DDMQ1-001139.0.raw Q1MicroLiteFM MISSING c38 √: 678 x: 1
0 INDEX 50*40
1 PARA 15*256
2 ENTER 15*255
3 TYPE 15*255
4 TYPE 15*255
5 TYPE
6 FORMAT 28*128
7 TEXT 9*462
8 TEXT 9*462
9 TEXT 9*462
10 DISK 15*255
11 FORM 15*255
12 CONTROL 67*37
13 PRINT 15*255
14 ATTACH 15*255
15 ITEM 67*37
16 NAME 16*232
17 NAME
18 NAME
19 NAME
20 NAME
21 CHANGE 15*255
22 FILES 15*255
23 TNE 15*255
24 TNE 15*255
25 FOOT 15*255
26 FOOT 15*255
27 FOOT 15*255
28 TRY 9*462
29 TRY
30 INSTR 9*462
31 INSTR 9*462
32 INSTR 9*462
33 INSTR 9*462
34 INSTR 9*462
35 LATHUND 9*462
36 IN 9*462
37 SORT 15*255
38 DTYPE
39 BREV1 9*462
DDMQ1-0011/000/DDMQ1-001140.0.raw Q1MicroLiteFM MISSING c38 √: 678 x: 1
DDMQ1-0011/000/DDMQ1-001141.0.raw Q1MicroLiteFM MISSING c38 √: 678 x: 1
DDMQ1-0011/000/DDMQ1-001142.0.raw Q1MicroLiteFM MISSING c38 √: 678 x: 1

FloppyTools (& AutoAutoArchaeologist) Repositories

https://github.com/Datamuseum-DK/FloppyTools

https://github.com/Datamuseum-DK/AutoArchaeologist