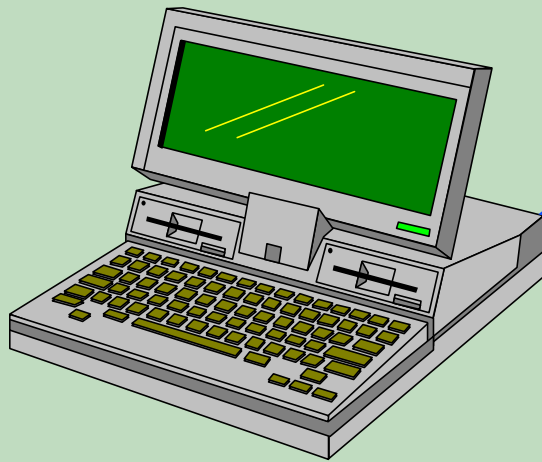


Software Design for Interfacing Märklin Digital



märklin
digital

Presented by:

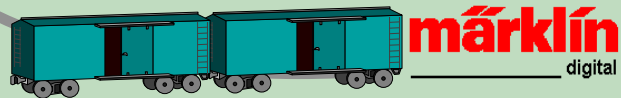
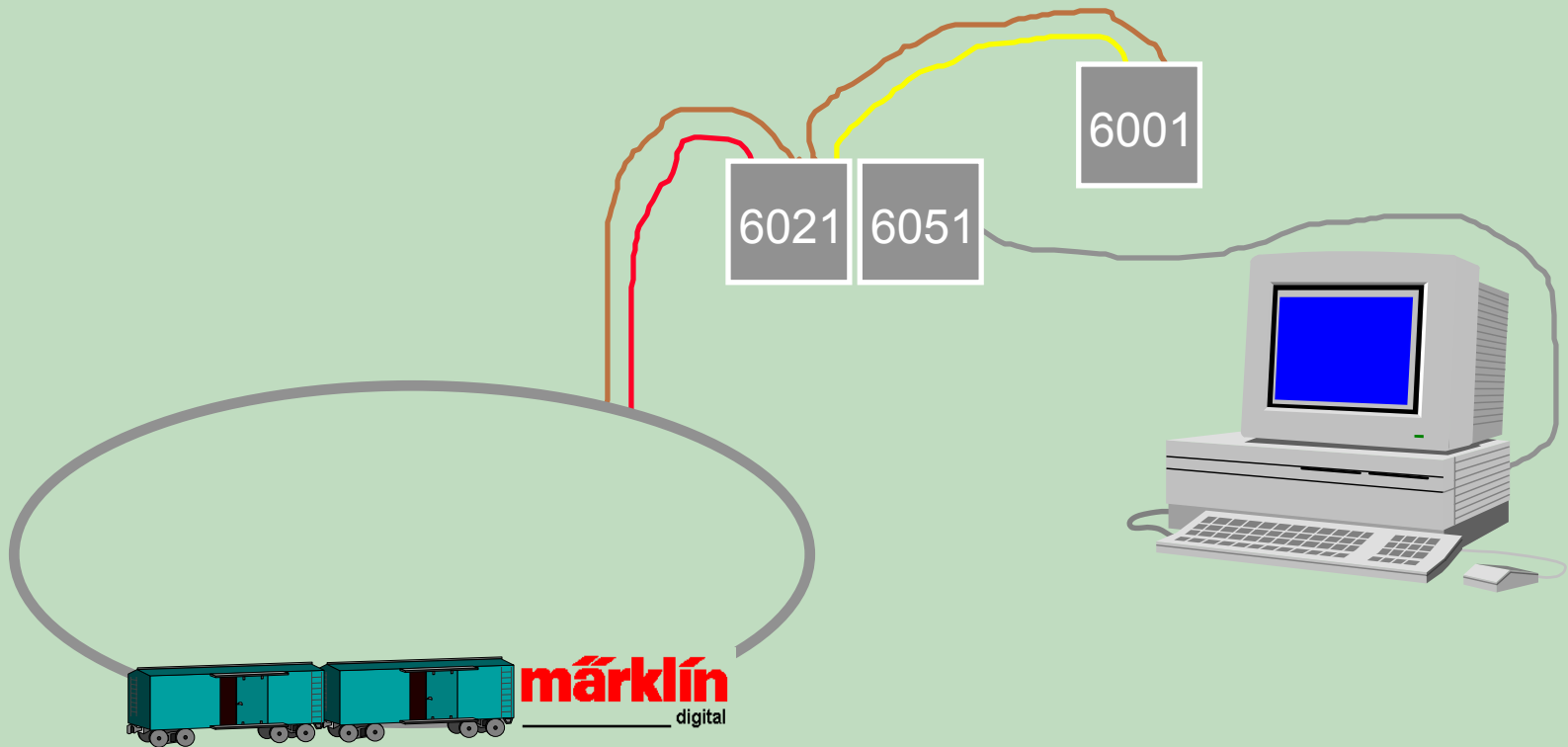
Robert Frowenfeld

märklin
digital

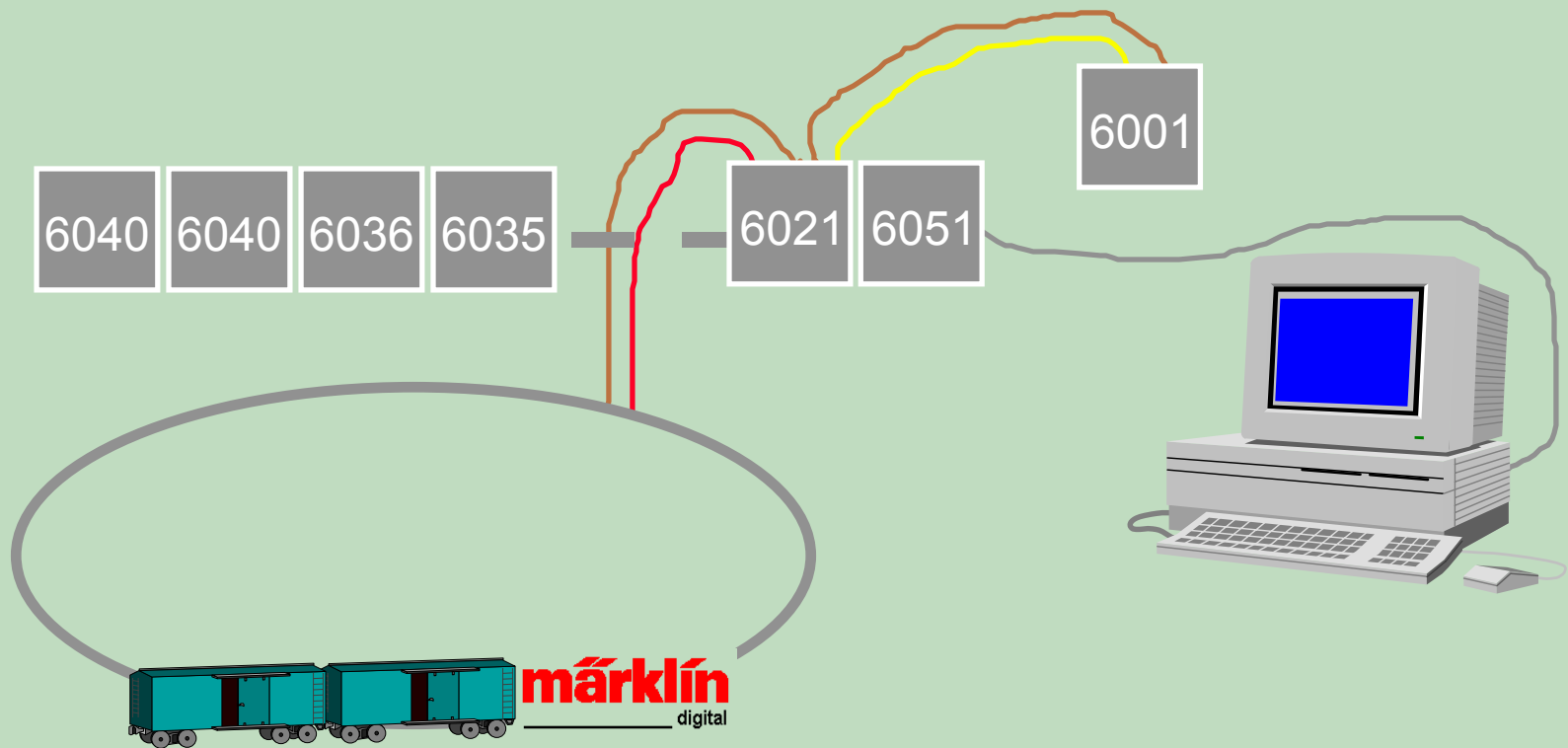
Presentation Outline

- The basics
- Programming languages
- DOS Vs Windows
- Interface commands
 - Open COM: port
 - Start / stop
 - Engines
 - Accessories (F1-F4)
 - Switches
 - s88 decoders
- Performance Issues
- Interface Idiosyncrasies
- The “lowly” 6023
- Visual controls
- Program considerations

The basics



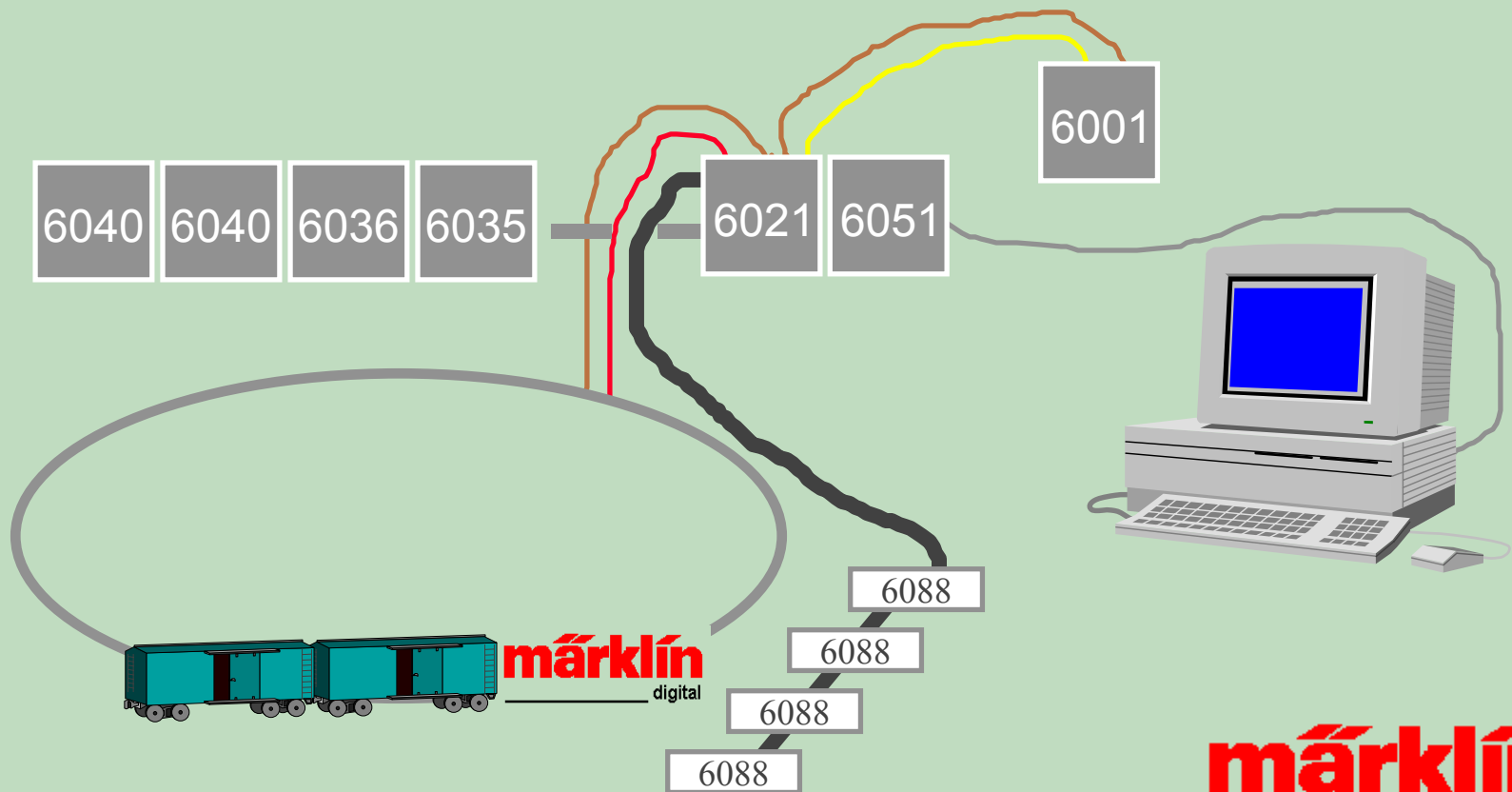
The basics



märklin
digital

märklin
digital

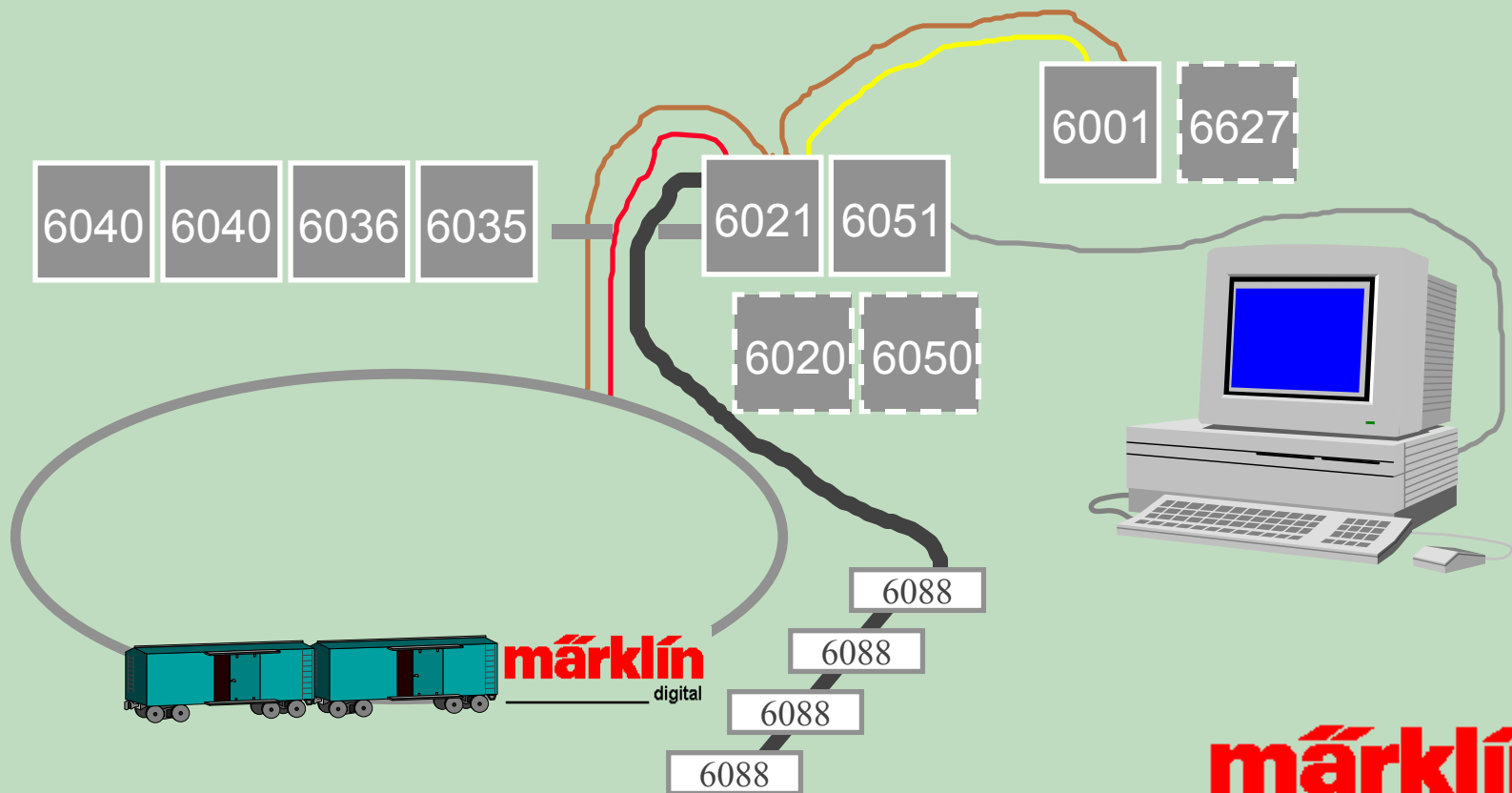
The basics



märklin
digital

märklin
digital

The basics



märklin
digital

märklin
digital

Programming Languages

- Basic
- Quick Basic (QB)
- Visual Basic (VB 2,3,4,5)
- Turbo Pascal
- Delphi

DOS Vs Windows

- Performance:
 - CPU Speed
 - RAM
 - Video
- Features you want to support:
 - Colors / graphics / mouse / sound / video
- Programming skills
- Availability of 3rd party tools

Interface Commands

- Open COM port

QuickBasic:

```
OPEN "COM1:2400,N,8,2,RS,DS,CD" FOR RANDOM AS #1
```

Visual Basic:

```
Comm1.Settings = "2400,n,8,2"
```

```
Comm1.CommPort = 1
```

```
Comm1.PortOpen = True
```

```
Comm1.Handshaking = If(RTS_Enabled%, 2, 0)
```

```
Comm1.RTSEnable = RTS_Enabled%
```

- Start / stop Central Unit

QuickBasic:

```
PRINT #1, CHR$(96); or PRINT #1, CHR$(97);
```

Visual Basic:

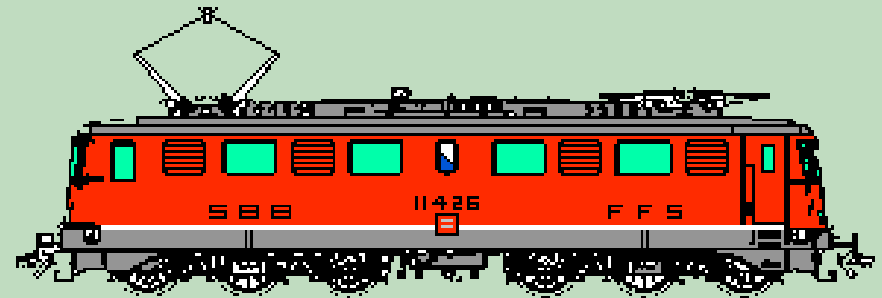
```
Comm1.Output = Chr$(96) or Comm1.Output = Chr$(97)
```

Interface Commands

● Engines

QuickBasic:

```
SPEED% = 6  
ENGINE% = 43  
FUNC% = -1 '0 = off, -1 = on  
PRINT #1, CHR$(SPEED% - 16 * FUNC%) + CHR$(ENGINE%);
```



Visual Basic:

```
Speed% = 6  
Engine% = 43  
Func% = -1 '0 = off, -1 = on  
Comm1.Output = Chr$(Speed% - 16 * Func%) + Chr$(Engine%)
```

OR

```
Call SendBytes (Speed% - 16 * Func%, Engine%)
```

```
Sub SendBytes (byte1%, byte2%)  
Comm1.Output = Chr$(byte1%)  
If byte2% > 0 Then  
    Comm1.Output = Chr$(byte2%)  
End If
```

Interface Commands

- Accessories

Rule:

$$64 + F1*1 + F2*2 + F3*4 + F4*8 \Rightarrow 64 - 79$$

Visual Basic:

Accessory% = 10 'panorama car

F(1) = -1 'move waiter forward

F(2) = 0 'move waiter backward

F(3) = -1 'car lights on

F(4) = -1 'restaurant lights on

Byte1% = 64

For X% = 1 To 4

 If F(X%) Then Byte1% = Byte1% + 2^(X% - 1)

Next

Byte2% = Accessory%

Call Sendbytes (Byte1%, Byte2%)



Interface Commands

- Switches / signals

QuickBasic:

GREEN% = 33

RED% = 34

.....

POSITION% = RED%

SWITCH% = 145

PRINT #1, CHR\$(POSITION%) + CHR\$(SWITCH%);

Visual Basic:

Green% = 33: Red% = 34

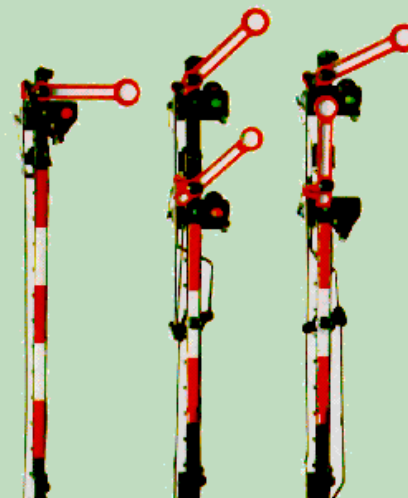
Position = Red%

Switch% = 145

Comm1.Output = Chr\$(Position%) + Chr\$(Switch%)

OR

Call SendBytes (Position%, Switch%)



Interface Commands

- s88 Track Detectors (destructive read):

QuickBasic:

```
DETECTORS% = 4  
PRINT #1, CHR$(128 + DETECTORS%);
```

Visual Basic:

```
Detectors% = 4  
Comm1.Output = Chr$(128 + Detectors%)
```

- s88 Track Detectors (non-destructive read):

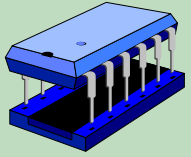
QuickBasic:

```
DETECTOR_NO% = 3  
PRINT #1, CHR$(192 + DETECTOR_NO%);
```

Visual Basic:

```
Detector_No% = 3  
Comm1.Output = Chr$(192 + Detector_No%)
```

Performance Issues

- CPU Speed
- Video (resolution, speed, RAM, drivers)
- RAM 
- Timing & polling
- Activity

Interface Idiosyncrasies

- Interface cable

The “lowly” 6023

- Introduced with Digital starter sets
- Works in ASCII and Binary
- Bi-directional
- Adjustable communications (baud) rate
- Limited inputs(4 s88's x 16 = 64)
- “Quirky”, inconsistent results
- No longer sold, limited support

Visual Controls

- VB Comm Control
(Crescent Software)
 - comes free with
Professional versions of VB



Programming Considerations

- Maintain status of all:
 - Engines
 - Solenoids
 - Contact tracks



THE END