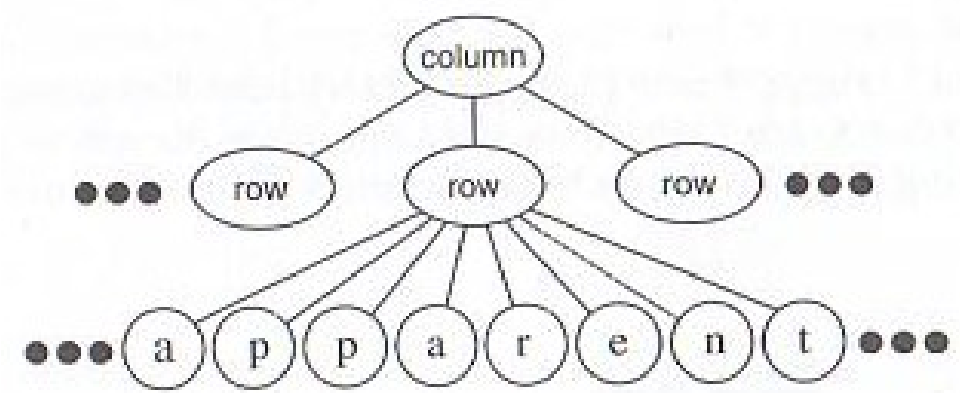
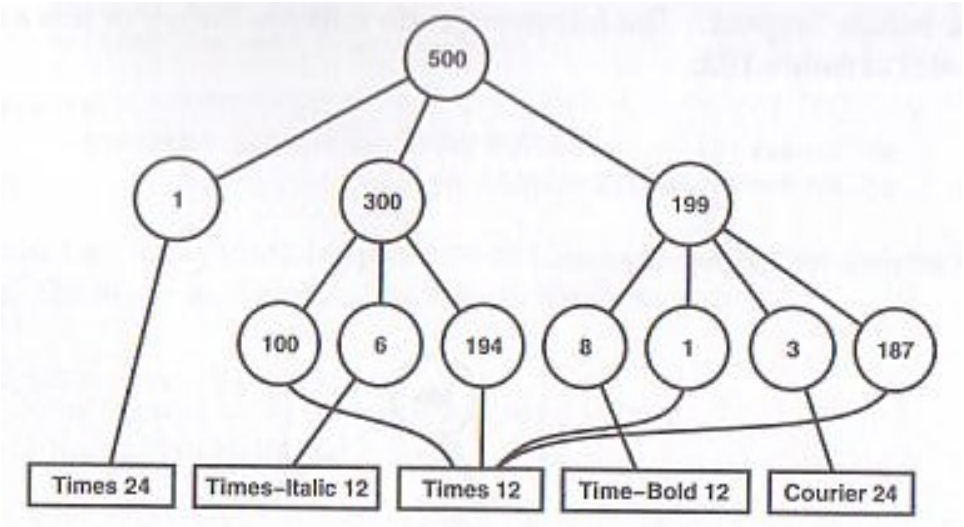
Plan for Today

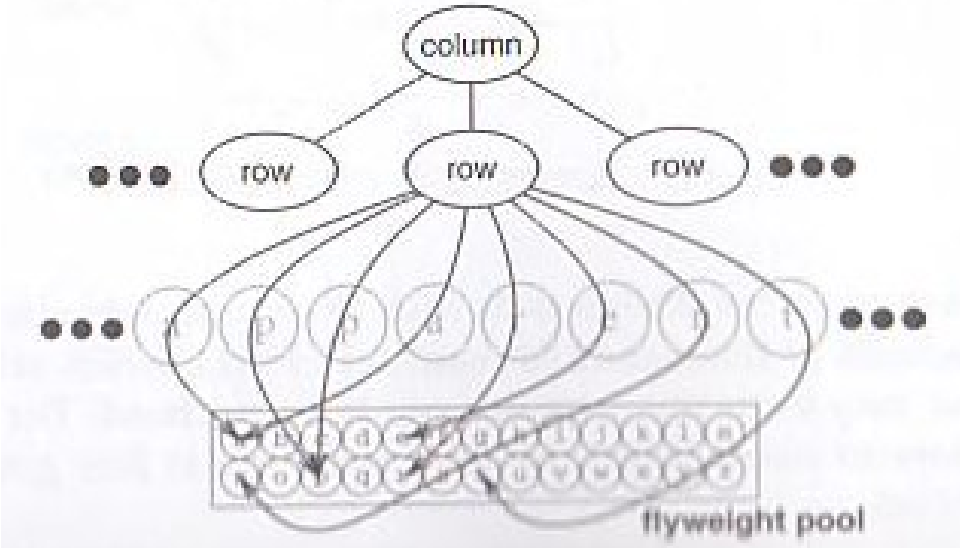
- Command pattern
- Template pattern
- Interpreter pattern

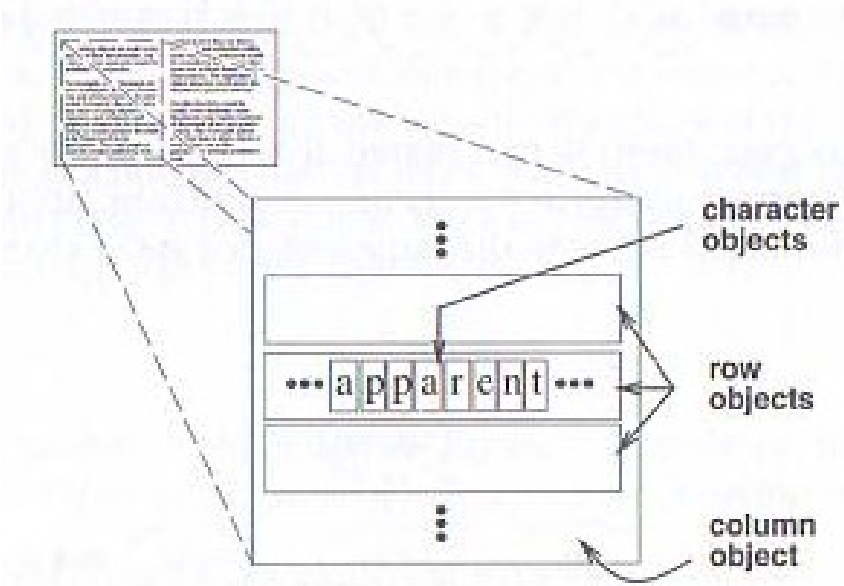
Lecture 51 – 3 December

Task 7 Questions?



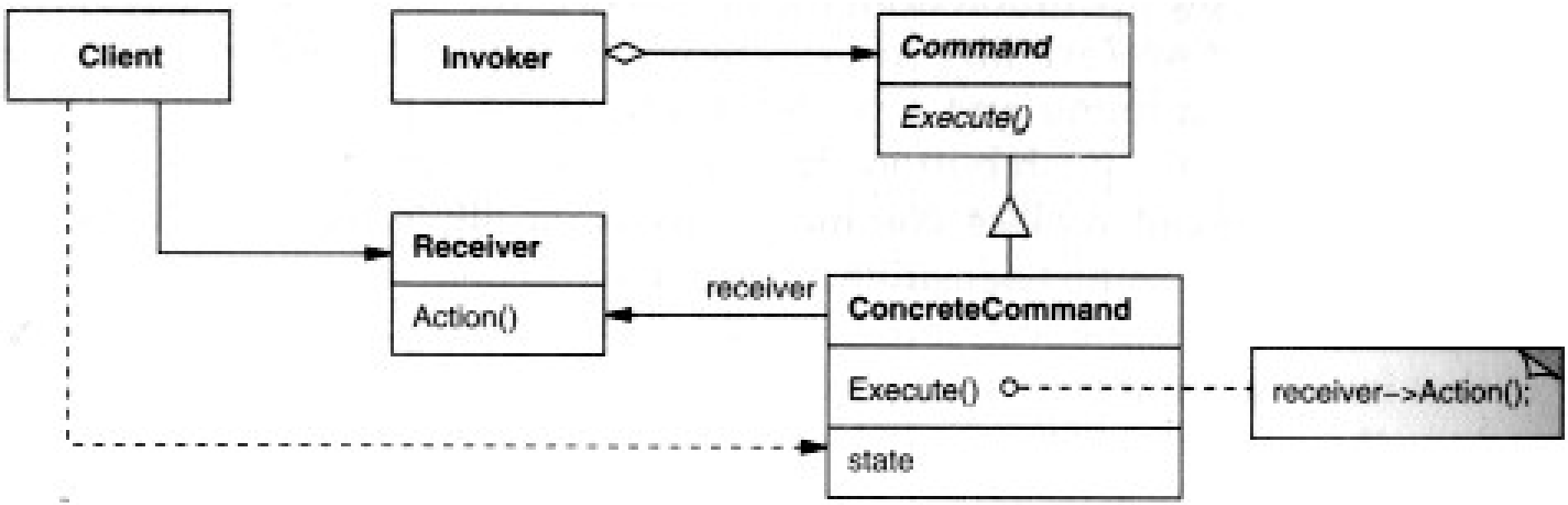






Command Pattern

- Encapsulates request as object, thereby allowing parameterizing of clients with different requests, queuing or logging requests, and supporting undoable operations
  - similar to message in Observer pattern
  - very useful for Interpreter pattern



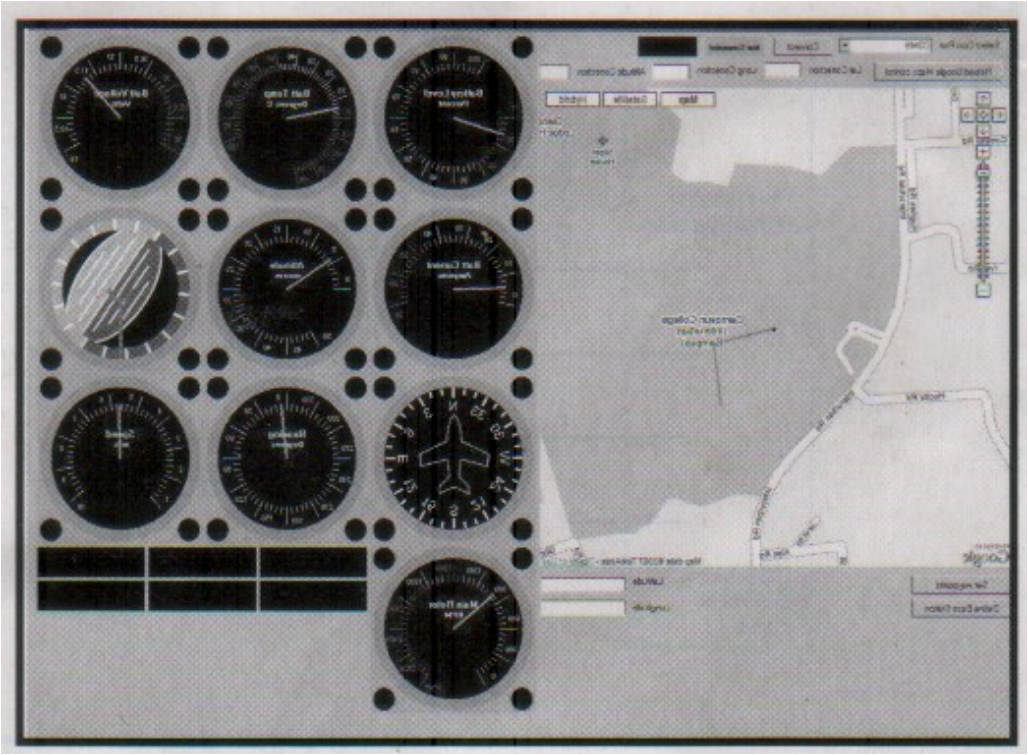
Command Pattern

- Applications
  - OOP variant of callbacks
  - specify, queue, and execute requests at different times
  - macros
  - scripts
  - undo / transaction rollback
  - redo
  - audit trail for testing

Command Pattern

Header		Length	Data			Checksum		Footer	
0xA5	0x5A	0x03	0x54	0x45	0x32	0x00	0xCE	0xCC	0x33
—	—	—	Family	Command	Payload	Sum of data and length		—	—

Table 1—This is an example of a packet sent between the base station and the helicopter. This packet, when received and decoded by the helicopter, will set the speed of the engine.



Group description	Group byte	Command	Command description
Testing/tuning	0x54	0x45	Engine speed adjust
	0x54	0x50	Pitch servo adjust
	0x54	0x52	Roll servo adjust
	0x54	0x43	Collective servo adjust
	0x54	0x51	Anti-torque servo adjust
	0x54	0x66	Set operations mode
	0x54	0xDD	General-purpose data dump
Flight operations	0x46	0x45	Engage engine
	0x46	0x48	Hover
	0x46	0x43	GPS Correction factor
	0x46	0x47	Go to GPS coordinates
	0x46	0x52	Return to base
	0x46	0x50	Request pre-flight packet
	0x46	0x4D	Discreet movement
	0x46	0x49	Request for information
Telemetry data	0x74	0x4C	Location
	0x74	0x48	Heading/speed/altitude
	0x74	0x5A	Attitude
	0x74	0x42	Battery status
	0x74	0x45	Error report
	0x74	0x50	Preflight packet
	0x74	0x52	Rotor RPM

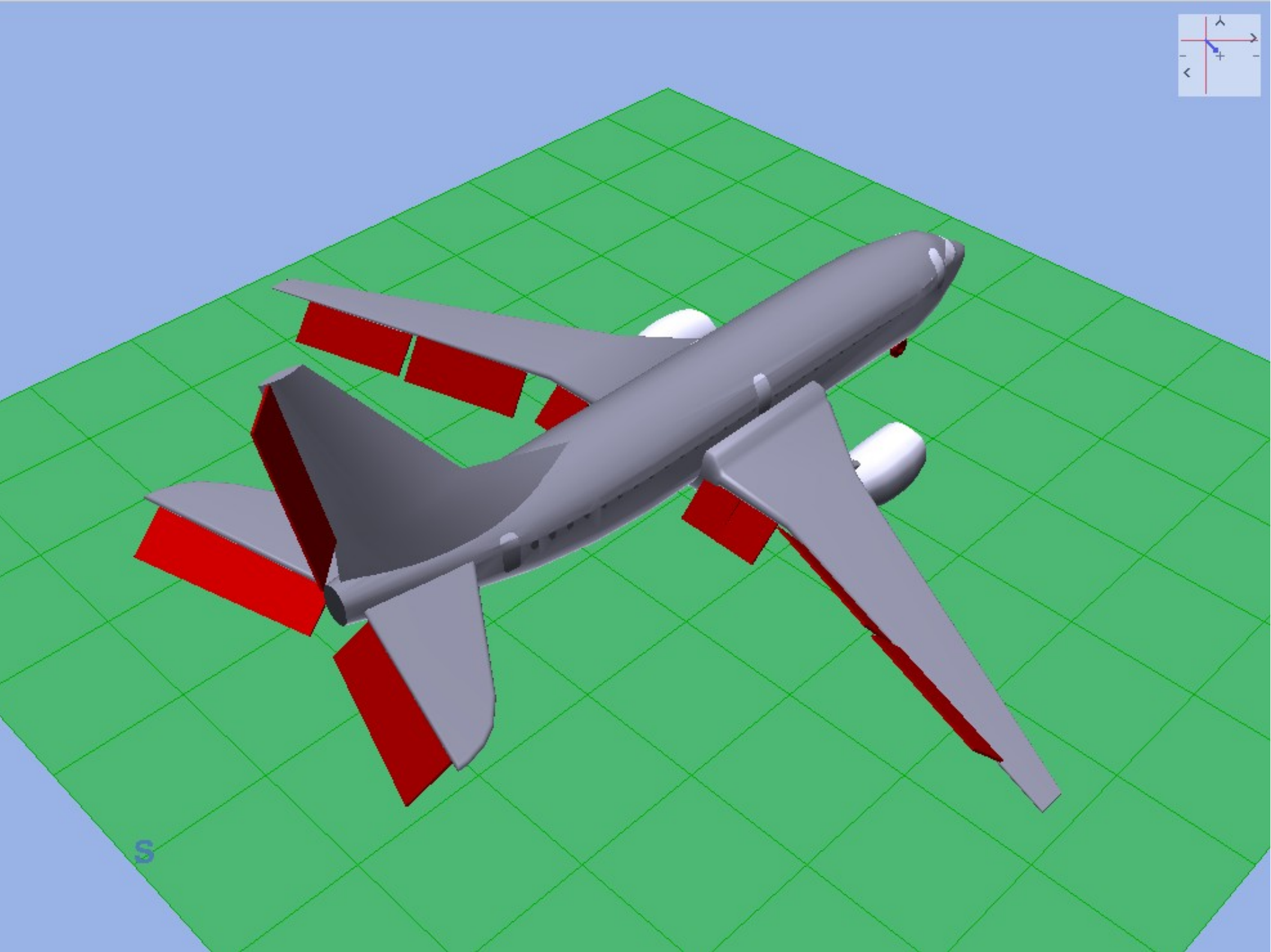
Table 2—This is a listing of the communications protocol used in our system. The protocol is flexible and it can expand to up to 65,535 commands, which would be broken into 256 groups. As an added bonus, the code is easily modified to accommodate the extra commands.

source: circuit cellar magazine

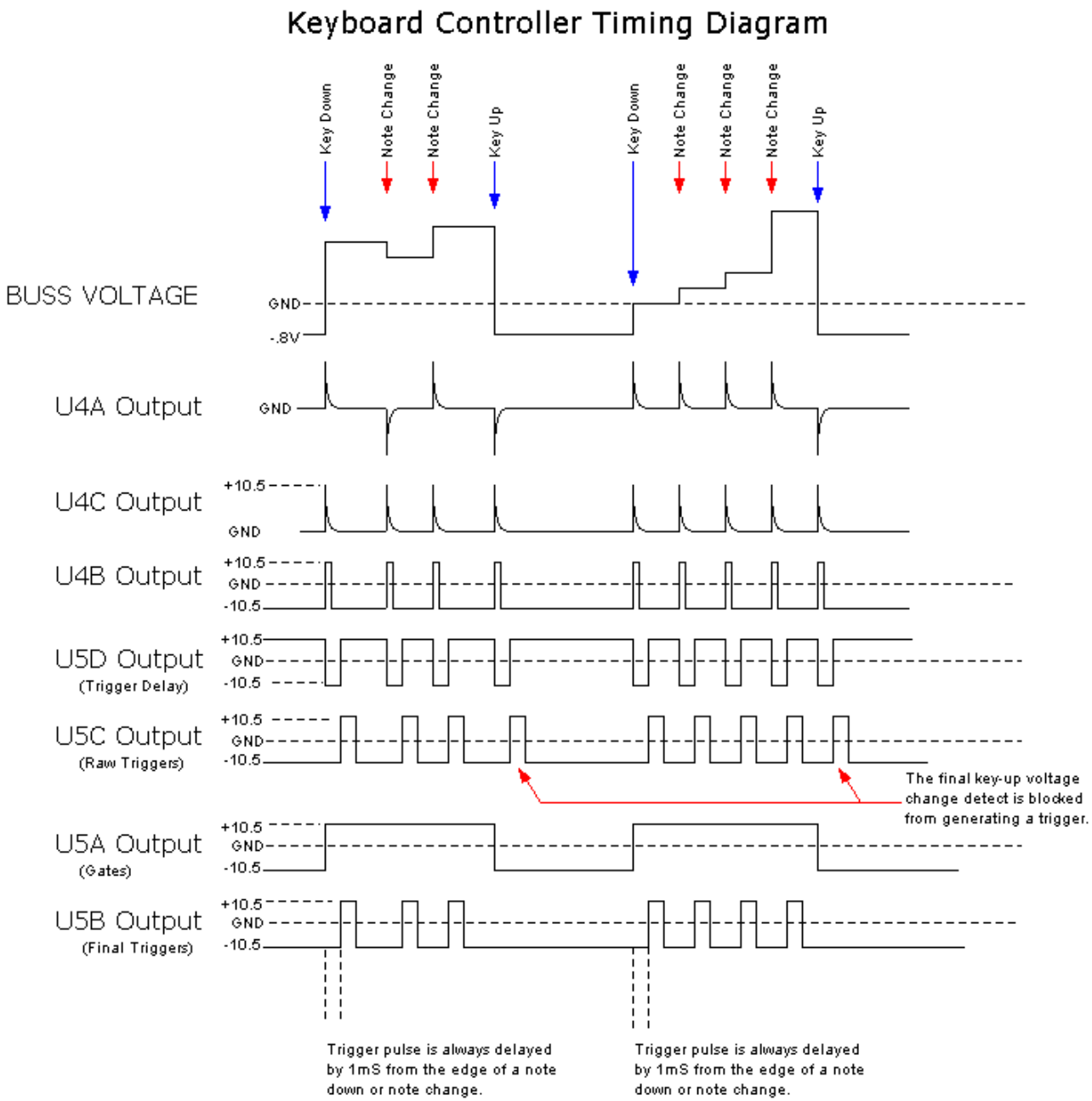

RESET
45°52'48.6"N 112°24'56.4"W
63°
-2160
SHLBY
DOG
WOOFY
KPIH
PWR
TOP
C
HDG
OBS
OBS
GS
NAV
NAV
--.-
2500
T H E
134.050 113.000
XCHG P S
190
P S
118.875 108.075
XCHG P S


NAV1 109.40 ↔ 110.30 ITPA
NAV2 116.70 111.40
Arriving at waypoint
DIS __.__NM BRG ___°
119.500 ↔ 122.800 COM1
126.450 121.700 COM2
120
110
100
95
90
80
70
GS 95KT
5
5
10
10
211°
21
24
15
12
30
33
6
3
W
E
LOC1
230
800
700
500
400
300
580
-900
4
2
2
4
NO GS
BARO MIN
230FT
29.92IN
18L
TRACK UP
TAMPA BAY
275
36L
60
600FT
589
27
580
RW18L
18R
574
TA OFF SCALE
5NM
OAT 0°C
XPDR1 2145 ALT R
LCL 09:05:01
INSET
SENSOR
PFD
OBS
CDI
XPDR
IDENT
TMR/REF
NRST
ALERTS


TRAFFIC
TERRAIN
200
10
10
5
5
5
5
180
170
160
150
140
2700
2600
2500
2400
2300
2
1
1
2
338°
HDG 338°
CRS 213°
TAS 160KT
STD BARO
NORTH UP
TFR: N/A
5NM
GPS
TERR
XTK 1214NM
OAT 0°C
XPDR 2145 ALT R
LCL 11:31:04

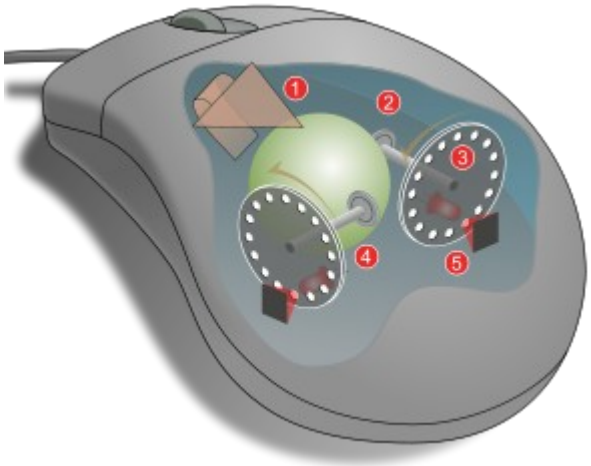

S

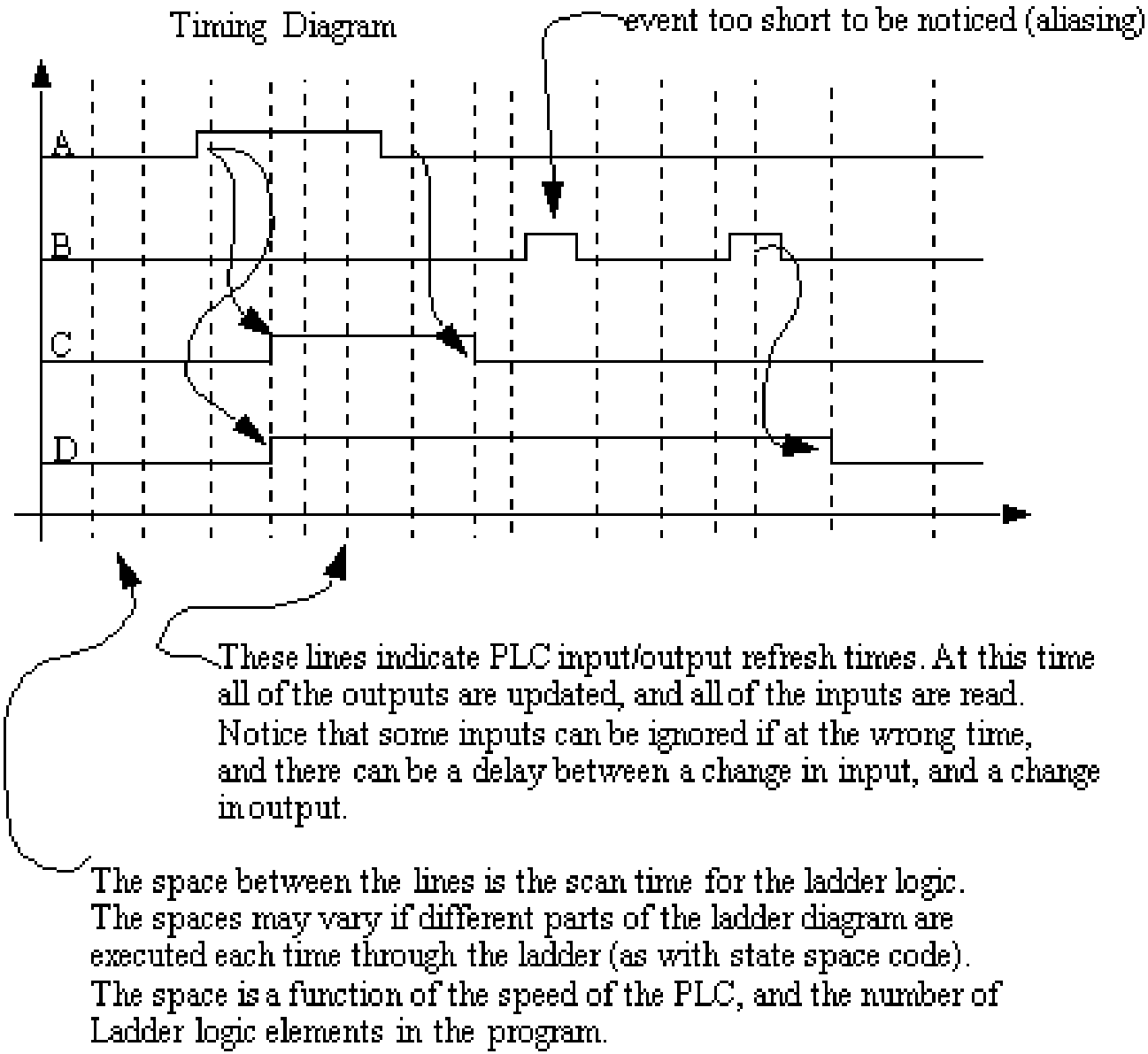
Command Pattern



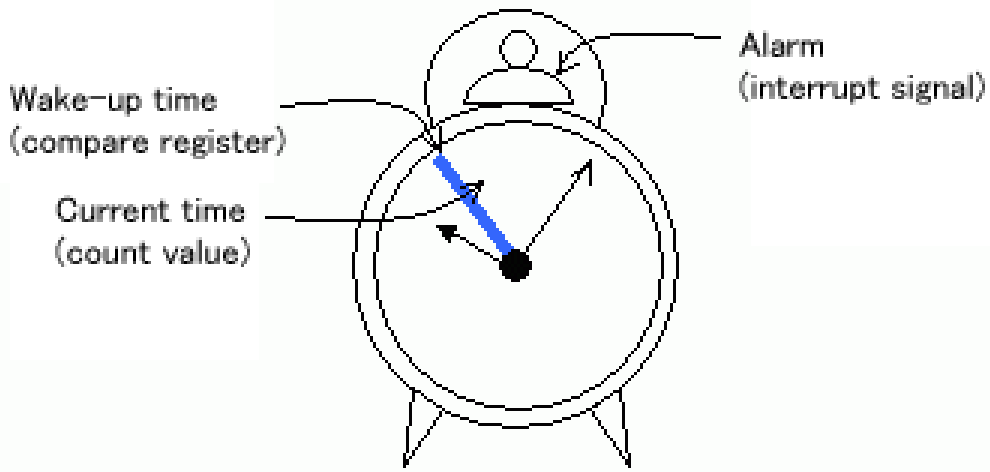
source: musicfromouterspace.com

Command Pattern

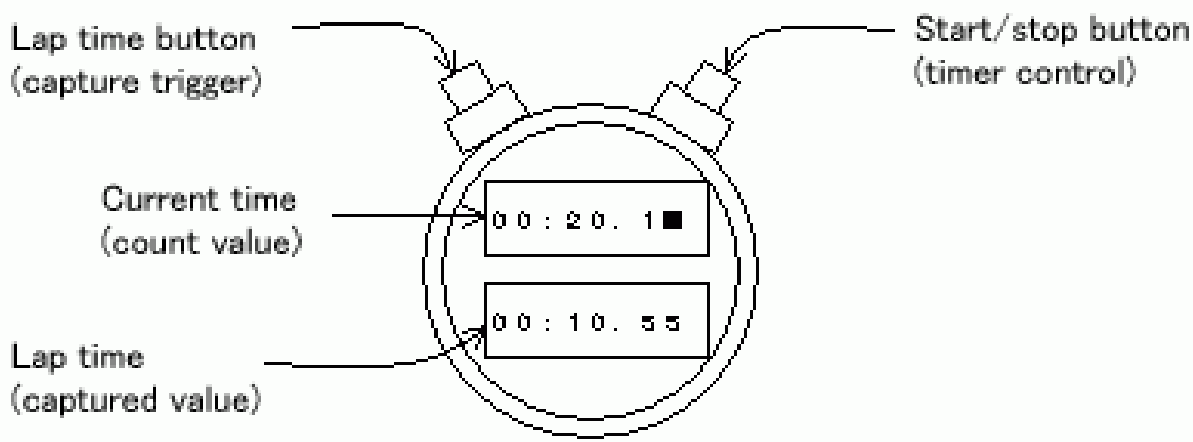




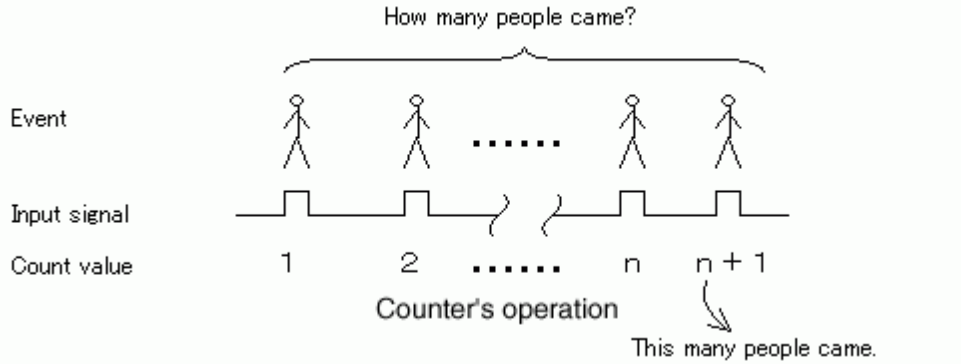
source: claymore.engineer.gvsu.edu, wikimedia.org

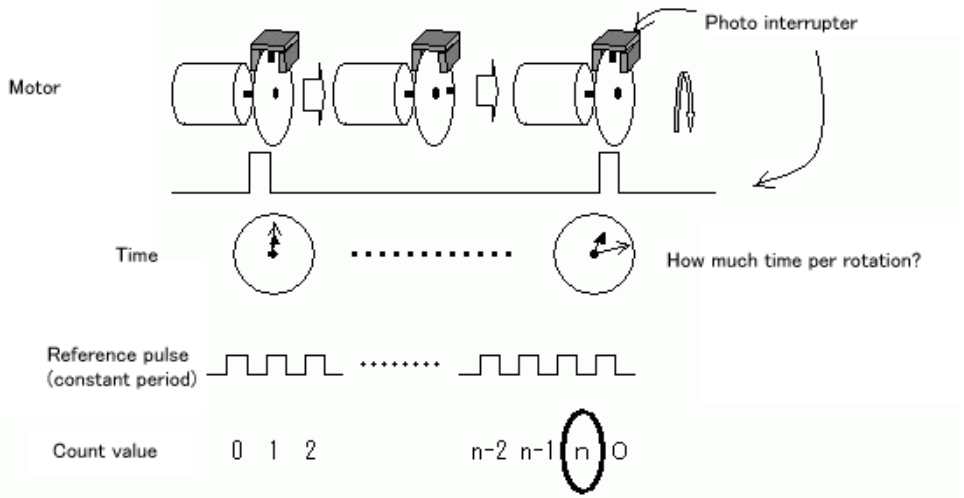


Alarm clock analogy for compare operation

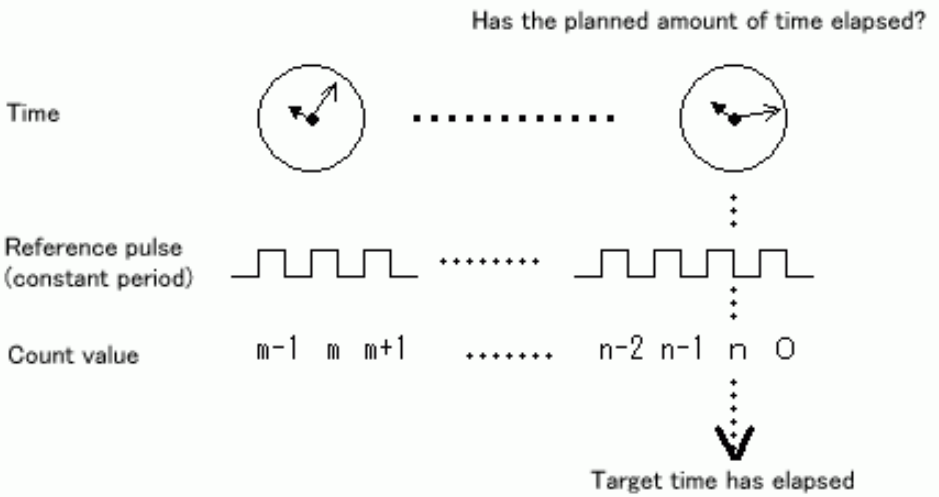


source: necel.com

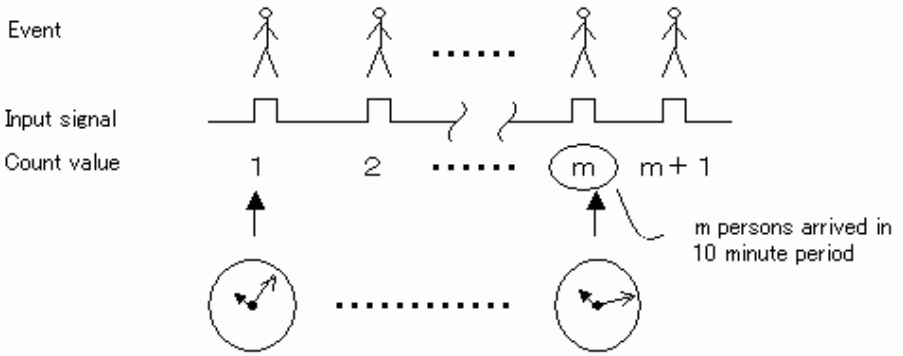




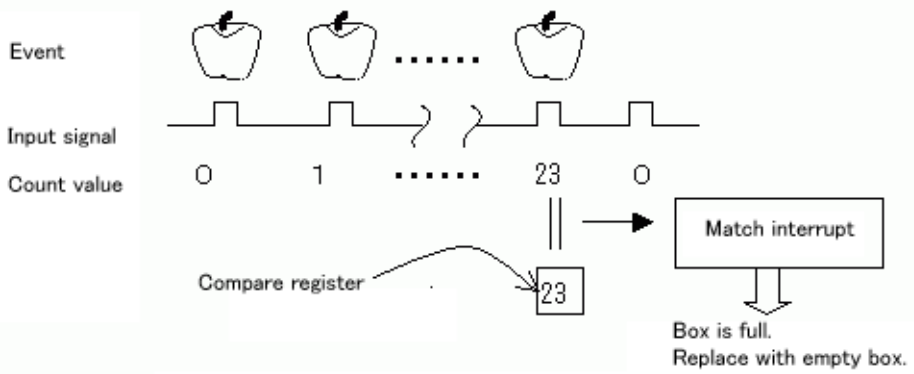
Time interval measurement example



Timer's operation

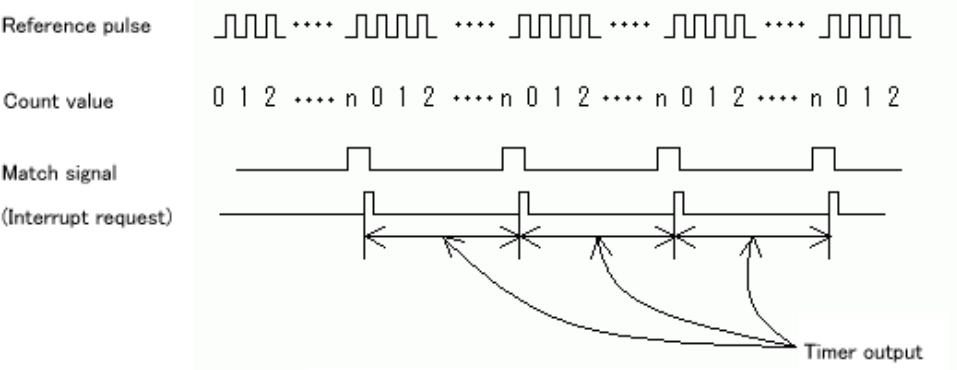


External event counter use example 1

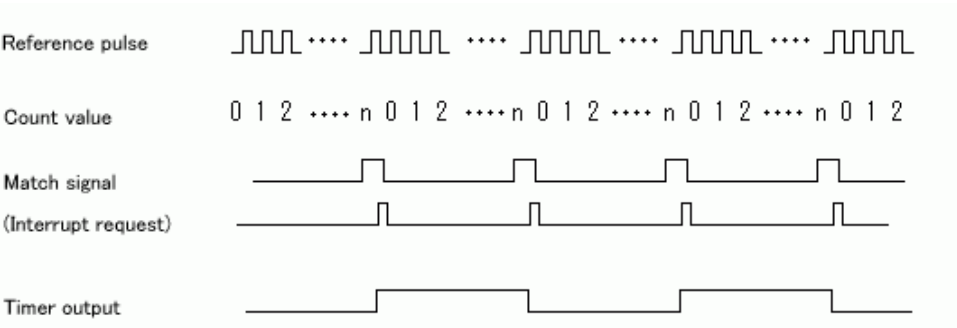


External event counter use example 2

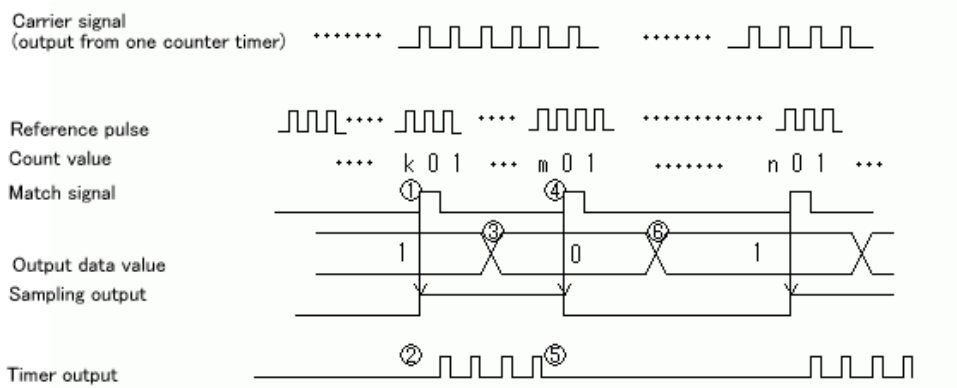
source: necel.com



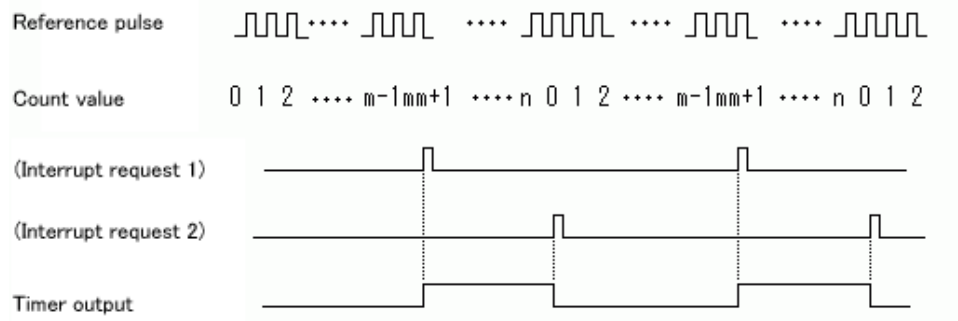
Pulse output operation example



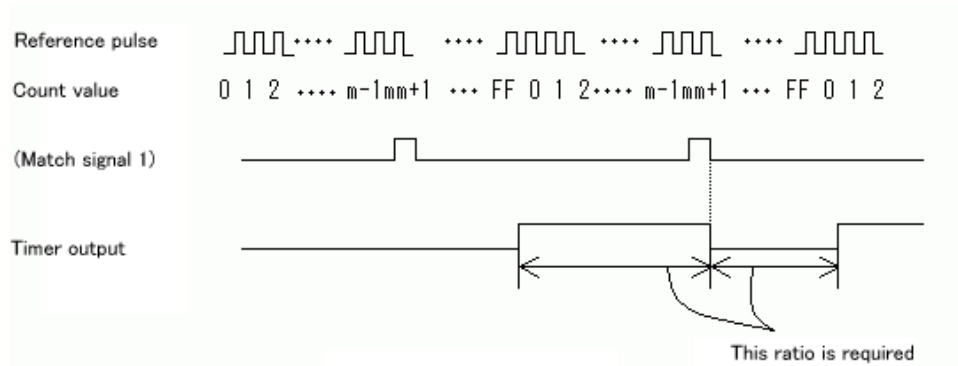
Square wave output operation example



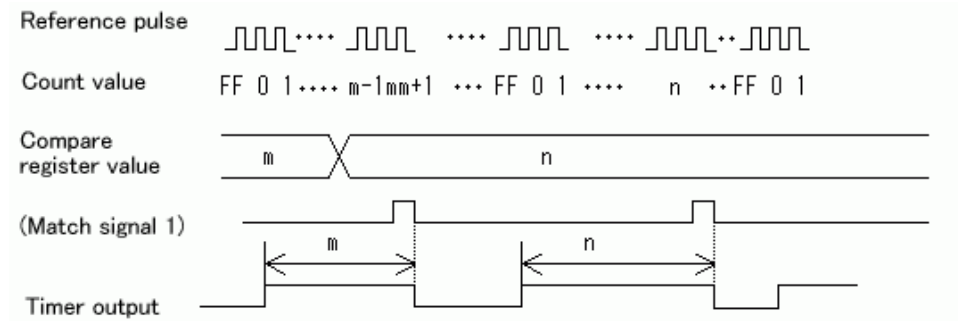
Carrier generation operation example



Pulse output operation example



PWM output operation example 1

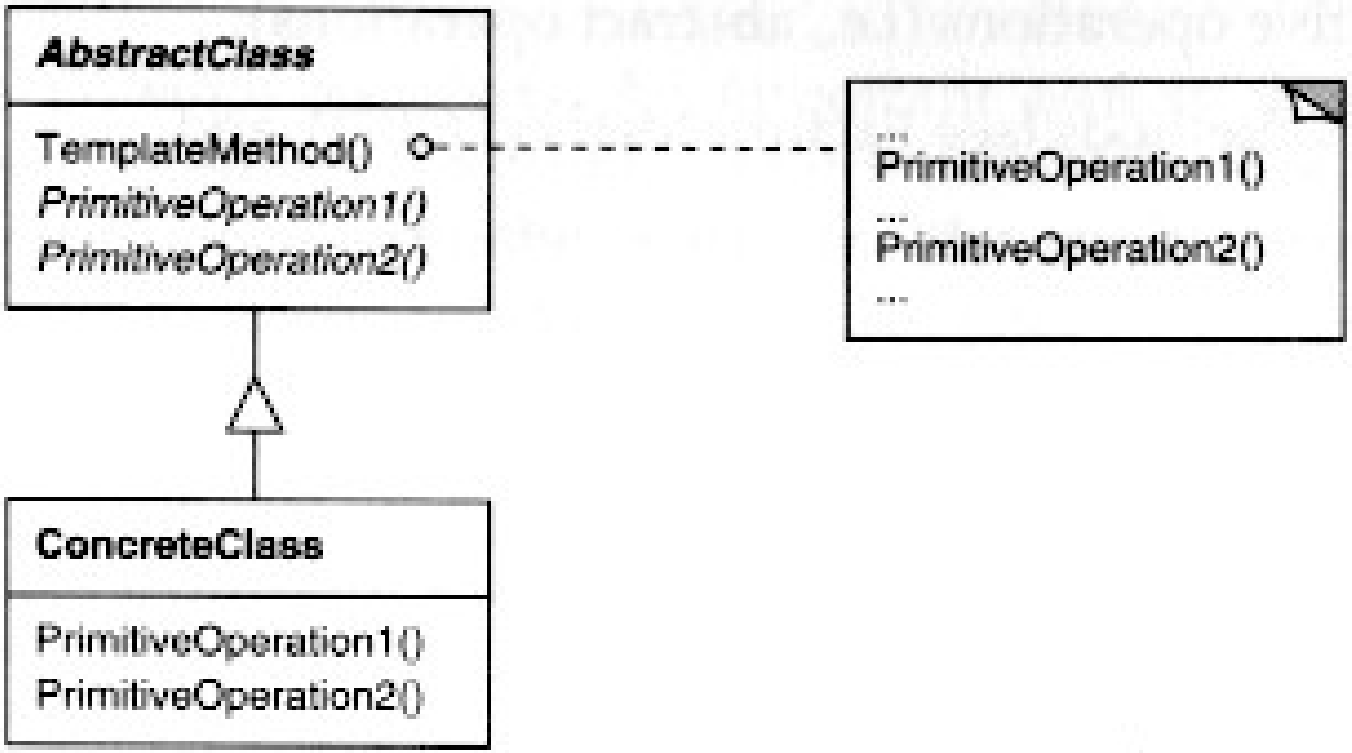


PWM output operation example 2

source: necel.com

Template Pattern

- Defines skeleton of algorithm in operation, deferring some steps to subclasses. Lets subclasses redefine certain steps of algorithm without changing its structure



Template Pattern

- Informal contract for staging calls to delegated implementations
  - analogous to Command pattern at code level
- Higher-level plug-and-play; e.g., introduction + body + conclusion
  - typical
    - contractually specify behavior on implementation
    - the plug-and-play *what* (at low "primitive" conceptual access level)
  - template
    - augment typical contract with (runtime) <u>agreement</u> on usage of behavior
    - *how* the *what* will be called (at higher "composite" conceptual access level):

```
1. _open
2. _initialize
3. _start
4. _run
5. _stop
6. _close
```

possibly some sharable elements; reduces code duplication

Template Pattern

- Similar to macro:

```
interface I_Viewer {
  // macros
  void _doOpenFile();
  void _doCloseFile();
  void _doExit();

  // callbacks / hooks
  void _handleAboutToOpen();
  void _handleOpening(double percentDone);
  void _handleDoneOpening();
}
```

```
interface I_Document {
  I_Document _requestDocument(I_Viewer viewer);
  void _doOpen();
  void _doRead();
  void _doRender();
  void _doClose();
}
```

```
public class Viewer implements I_Viewer {
  public void doOpenFile() {
    I_Document document = Document._requestDocument(this); // filename?
      document._doOpen();    // minimize handling of Document data
      document._doRead();
      document._doRender();
      document._doClose();
  }
  ...
  public void _handleAboutToOpen()                { ... } // optional reactions
  public void _handleOpening(double percentDone) { ... }
  public void _handleDoneOpening()                { ... }
}
```