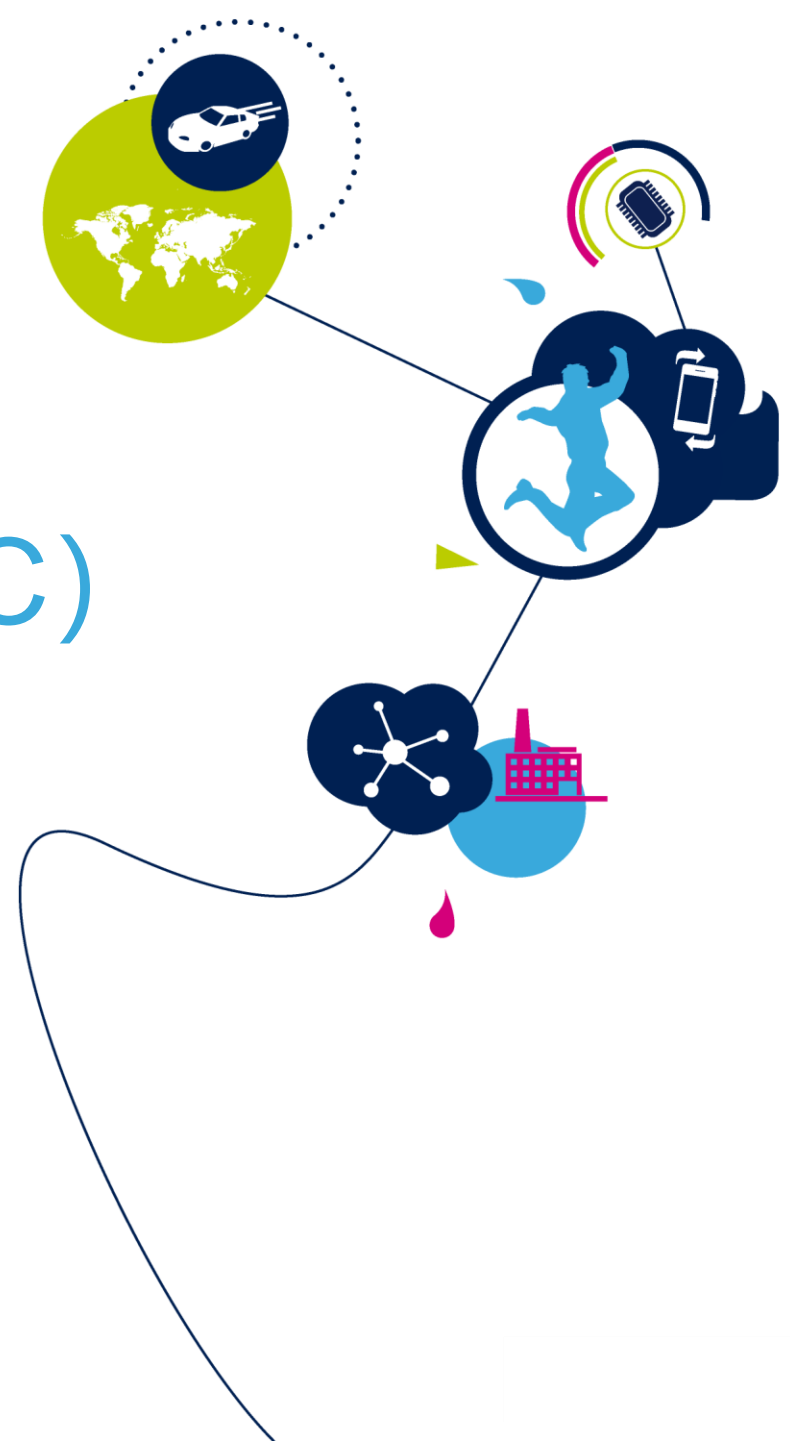


ST 可编程传感器 (FSM+MLC)

Sep/2019



可编程 FSM

最多16程序可同时运行

一套完整的命令和条件指令集

每个程序都可以输出多个状态

编程基于加速度，陀螺，磁力计的处理。



LSM6DSO/OQ/OX 提供了灵活的可编程的姿态识别算法

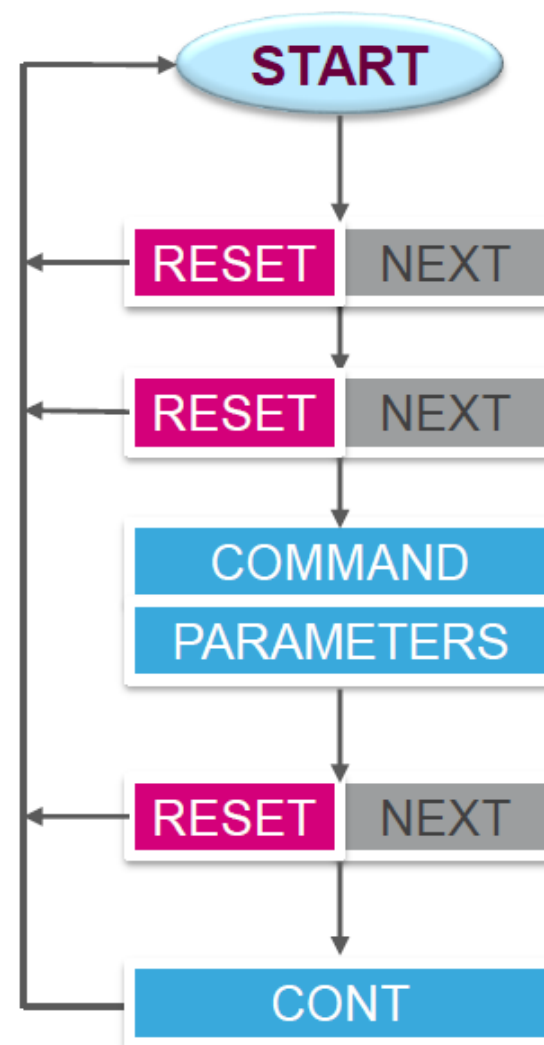
状态机是由一系列的命令和条件组成的逻辑，执行器按照这个逻辑和传感器数据生成的周期一步一步执行。

命令：

- 每个命令占8个比特，有时后面还有参数。
- 当执行器执行的到一个命令时，指针会立刻指向下一个命令或条件并进行评估。

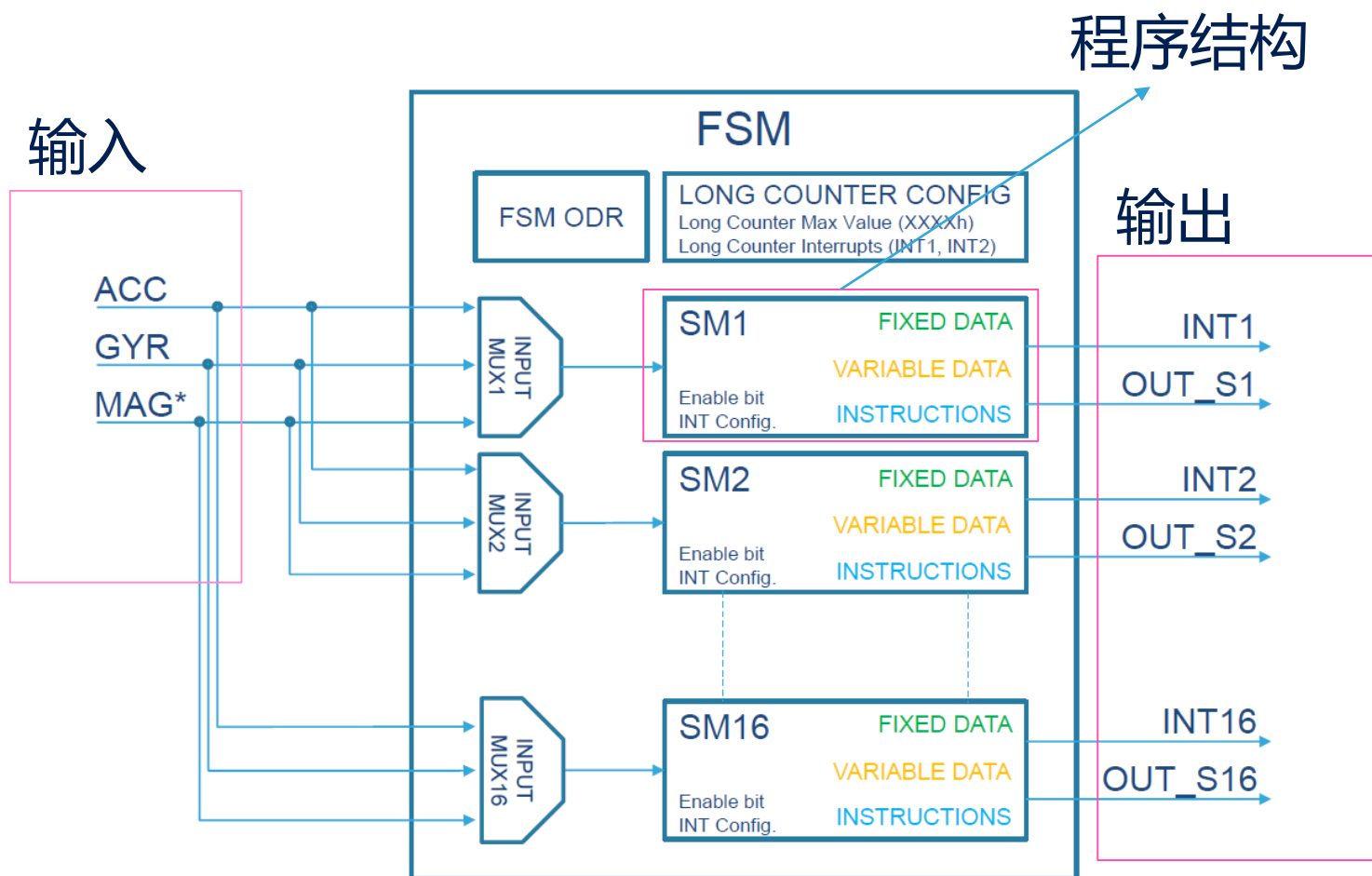
条件：

- 条件是由4bits 的复位和4bits继续执行的两个条件组成。
- 当复位条件为真时，程序指针会指向上一次配置的复位地址，复位地址可以有SRP和CRP两个指令实时指定。
- 当继续执行条件为真时，程序指针会指向下一行命令或条件，并当新的传感器数据产生时，执行器立即执行新的命令或条件。
- 如果条件语句是NONE时，指针不会变，当下一次的传感器数据产生时执行器还是评估原本这一行的条件。



状态机整体结构

4



总共由16个可编程的状态机组成，且每个状态机的是独立运行的基于相同的输入。

输入：

通过SIMNUX(0X23)命令可以选择输入信号

- 000 加速度数据 [ax ay az av]
- 001 陀螺仪数据 [gx gy gz gv]
- 010 校准后的磁力计数据[mx my mz mv]
- 111 陀螺的积分角度值 [dx dy dz dv]

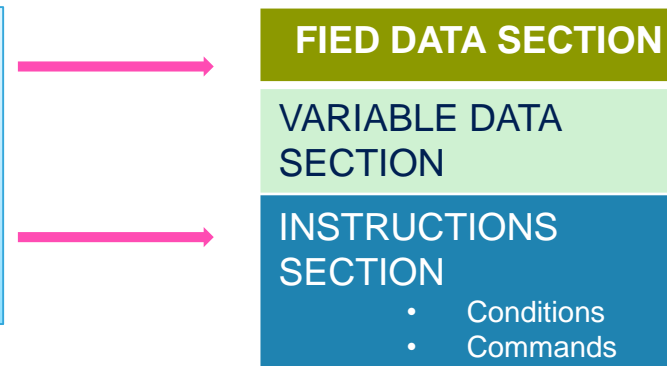
以上每类输入的v表示 三轴的矢量 $V = \sqrt{X^2 + Y^2 + Z^2}$

输出：

- 通过一些指令可以把状态机执行的状态输出到out 寄存器，每一个状态机都有一个对应的输出寄存器。
- 每一个状态机都可以选择在中断管脚 INT1，INT2 产生对应的中断。

一个简单程序可以由一下组成

- 一个数据区域，包括一个固定数据组成的固定去和一个变量组成的数据区。
- 指令区， 有条件 and 命令语句组成。



固定数据区是由六个字节组成，位于每个状态机的最前面，用于指定哪些资源指令区域将要用到的以及一些有用的参数。

变量区是一个可变字节的区域，位于固定区的后面，用于分配变量的区域

指令区也是一个可变字节的区域，位于变量区的后面，用户通过命令和条件定义算法的一个区域。

三个区域最大长度不超过256个字节。

	NAME	7	6	5	4	3	2	1	0
0	CONFIG A	NR_THRESH(1:0)		NR_MASK(1:0)		NR_LTIMER(1:0)		NR_TIMER(1:0)	
1	CONFIG B	DES	HYST	ANGLE	PAS	N/A	STOPDONE	LC	JMP
2	SIZE	PROGRAM SIZE(7:0)							
3	SETTINGS	MASKEL(1:0)		SIGNED	R_TAM	THRS3SEL	IN_SEL(2:0)		
4	RESET POINTER	RESET POINTER(7:0)							
5	PROGRAM POINTER	PROGRAM POINTER(7:0)							

可能用到资源：

- 最多三个不同的阈值
- 一个回滞变量
- 最多四个不同的时间变量
- 最多三个mask 变量
- 一个降频器
- 等等...

算法资源： 由用户定义

程序大小和指针定义： 这个必须预先定义以保证状态机的正常运行

其它：不是由用户指定，而是有指令区特定指令自动管理的。

注意：状态机的长度必须是偶数

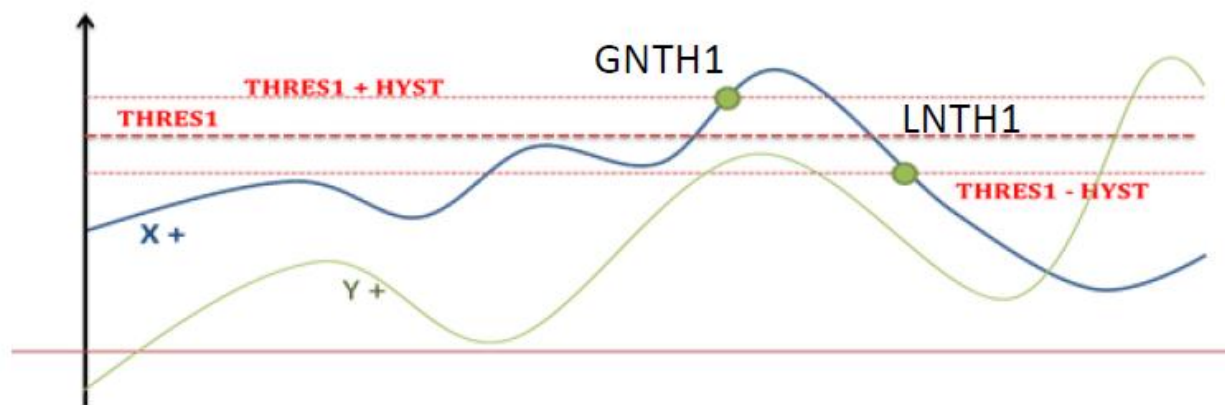
阈值：

它是用于在指定输入数据进行比较的判断条件

- 一些条件指令会指定特定的阈值 (GNTH1, LHTH1, GNTH2, 等等)
- 一些命令指令可以实时修改这些阈值。

回滞变量：用户也可以指定回滞变量，它可以自动在阈值变量上加或减去这个值。

- 当用于大于条件判断时阈值会自动加上这个变量
- 当用于小于条件判断时阈值会自动减去这个变量



Masks:

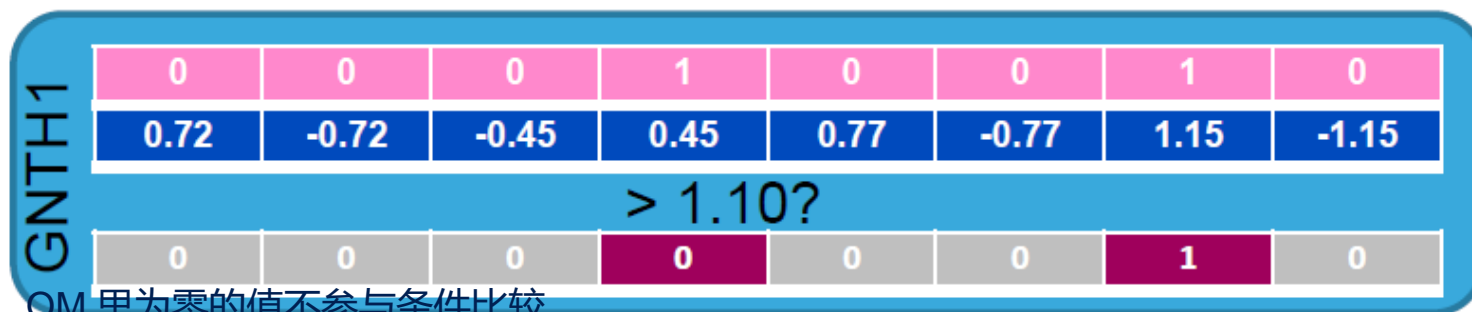
用于指定输入数据的那个轴进行阈值条件以及过零的判断。

首先mask 分为初始mask(OM)和零时mask(TM).在初始时他们是相等的

OM +X	OM -X	OM +Y	OM -Y	OM +Z	OM -Z	OM +V	OM -V	OM
TM +X	TM -X	TM +Y	TM -Y	TM +Z	TM -Z	TM +V	TM -V	TM

示例:

- GNTH1 条件指令 (阈值1=1.10g , OM=12h);
- 当前加速度输入为[0.72 -0.45 0.77 1.15]



OM
加速度
阈值
TM

OM 里为零的值不参与条件比较

TM: 零时 mask可以输出到output 寄存器

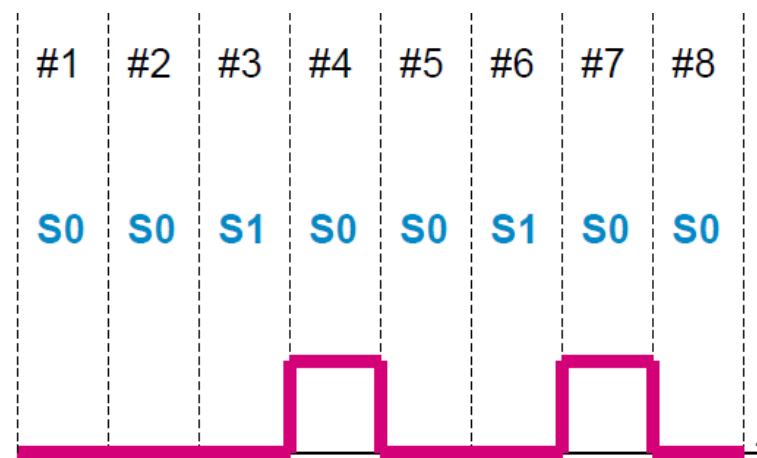
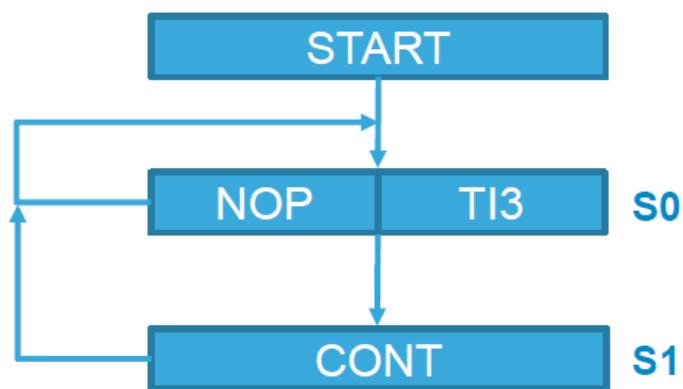
时间变量:

用于对事件持续事件的判断。

- 最多两个短时钟定时器 (T13, T14)
- 最多两个长时钟定时器 (T11, T12)
- 特定的条件指令选择特定的定时器 (T11, T12, T13, T14)
- 选用专门的指令实时修改定时器的值 (stimer3, stimer4 ,等的)

定时器的值在触发之前是随着处理的传感器数据变化的, 如来一组新的采样数据, 定时器就自动加1。

示例: (T13=0X03)



降频器:

可用来降低状态机处理传感器数据的频率，状态机原本是按特定数据输出频率处理数据的，可以通过修改降频器的因子（DEST）进一步降低状态机数据处理的频率。

- DEST 存储了降频的因子
- DESC 是一个递减的时钟定时器
- 当DESC 变为零时，状态机才开始处理数据，并且把DEST里的降频因子重新写入DESC。
- 每个状态机都有自己独有的降频器。

陀螺的积分角度:

(DELTAT,DX/Y/Z/V)用来做陀螺的角度积分。

- DELTAT 存放陀螺的采样时间。
- DX/Y/Z/V存放陀螺的积分角度
- Masks和阈值都可应用于算法中相关的角度。

整个数据区

12

数据区可以包含最多字节为：

- 固定数据区 (6字节)
- 3个阈值 (6字节)
- 回滞变量 (2字节)
- 3个masks(6字节)
- 陀螺积分 (10字节)
- 时钟计数器 (1/2 字节)
- 2个长时钟定时器 (4字节)
- 2个短时钟定时器 (2字节)
- 减频器
- 前一个轴的状态 (1字节)

指令区紧接着数据区

FIXED
VARIABLE
INSTRUCTION

		7	6	5	4	3	2	1	0
0	CONFIG A	NR_THRESH(1:0)		NR_MASK(1:0)		NR_LTIMER(1:0)		NR_TIMER(1:0)	
1	CONFIG B	DES	HYST	ANGLE	PAS	N/A	STOPDONE	LC	JMP
2	SIZE	PROGRAM SIZE(7:0)							
3	SETTINGS	MASKEL(1:0)		SIGNED	R_TAM	THRS3SEL	INSEL(2:0)		
4	RP	RESET POINTER(7:0)							
5	PP	PROGRAM POINTER(7:0)							
6	THRESH1	THRESH1(15:0)							
7									
8	THRESH2	THRESH2(15:0)							
9									
10	THRESH3	THRESH3(15:0)							
11									
12	HYST	HYSTERESIS(15:0)							
13									
14	MASKA	MASKA(7:0)							
15	TMASKA	TMASKA(7:0)							
16	MASKB	MASKB(7:0)							
17	TMASKB	TMASKB(7:0)							
18	MASKC	MASKC(7:0)							
19	TMASKC	TMASKC(7:0)							
20	DELTAT	DELTAT(15:0)							
21									
22	DX	DX(15:0)							
23									
24	DY	DY(15:0)							
25									
26	DZ	DZ(15:0)							
27									
28	DV	DV(15:0)							
29									
30	TC	TC(15:0) or TC(7:0)							
31									
32	TIMER1	TIMER1(15:0)							
33									
34	TIMER2	TIMER2(15:0)							
35									
36	TIMER3	TIMER3(7:0)							
37	TIMER4	TIMER4(7:0)							
38	DEST	DEST(7:0)							
39	DESC	DESC(7:0)							
40	PAS	SCTC	CANGLE	MSKIT	MSKITEQ	SIGN X	SIGN Y	SIGN Z	SIGN V

COMMAND	DESCRIPTION		PARAMETERS
0X00	STOP	stop execution, and wait new start from reset-point	None
0X11	CONT	Continues execution from reset-point	None
0X22	CONTREL	Continues execution from reset-point resetting temporary mask	None
0X33	SRP	Set reset-point to first program line	None
0X44	CRP	Clear reset-point to frist program line	None
0X55	SETP	Set parameter in program memory	byte 1 address, byte 2 value
0X66	SELMA	Select MASKA and TMASKA as current mask	None
0X77	SELMB	Select MASKB and TMASKB as current mask	None
0X88	SELMC	Select MASKC and TMASKC as current mask	None
0X99	OUTC	Write the temporary mask to output registers	None
0XAA	STHR1	Set new value to THRESH1 register	byte 1 and byte 2 value threshold 1 register
0XBB	STHR2	Set new value to THRESH2 register	byte 1 and byte 2 value threshold 2 register
0XCC	SELTHR1	Selects THRESH1 instead of THRESH3	None
0XDD	SELTHR3	Selects THRESH3 instead of THRESH1	None
0XEE	SISW	Swaps sign information to opposite in selected mask	None
0XFF	REL	Rest temporary mask to default	None
0X12	SSIGN0	Set UNSIGNED comparison mode	None
0X13	SSIGN1	Set SIGNED comparison mode	None
0X14	SRTAM0	Do not reset temporary mask after a next condition true	None
0X21	SRTAM1	Rest temporary mask after a next condition trure	None
0X23	SINMUX	Set input multiplexer	byte 1 value for input multiplexer
0X24	STIMER3	Set new value to TIMER3 register	byte 1 value timer3 register
0X31	STIMER4	Set new value to TIMER4 register	byte 1 valure timer4 register
0X32	SWAPMSK	Swap mask selection MASKA <=> MASKB; MASKC unaffected	None
0X34	INCR	Increase long counter +1 , check timeout and clear	None
0X41	JMP	Jump address for two Next conditions	byte 1 conditions , byte 2 and 3 jump address
0X42	CANGLE	Clear angle	None
0X43	SMA	Set MASKA and TMASKA	byte 1 value mask
0XDF	SMB	Set MASKB and TMASKB	byte 1 value mask
0XFE	SMC	Set MASKC and TMASKC	byte 1 value mask
0X5B	SCTC0	Clear Time Counter TC on next condition true (default one)	None
0X7C	SCTC1	Do not clear Time Counter TC on next condition true	None
0XC7	UMSKIT	Unmask interrupt generation when setting OUTS	None
0XEF	MSKITEQ	Mask interrupt generation when setting OUTS if outs is not changing	None
0XF5	MSKITEQ	Mask interrupt generation when setting OUTS	None

命令指令

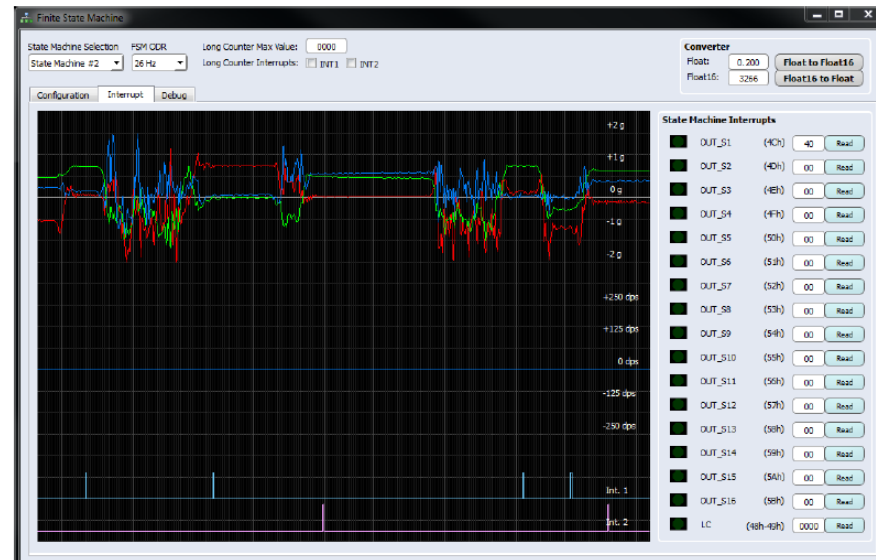
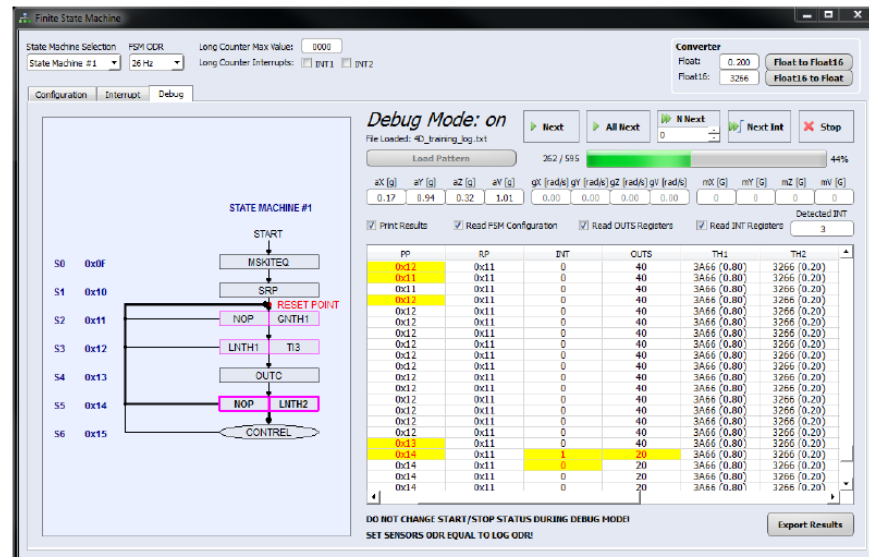
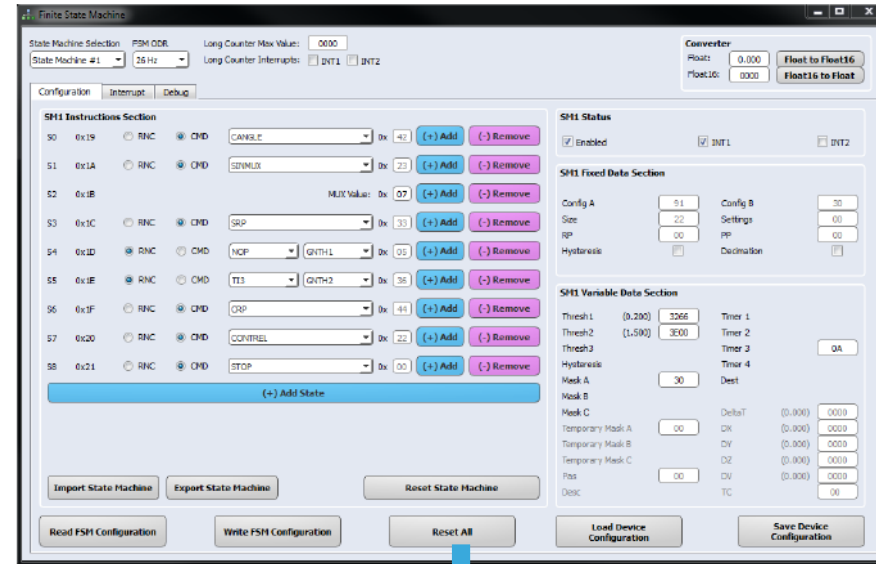
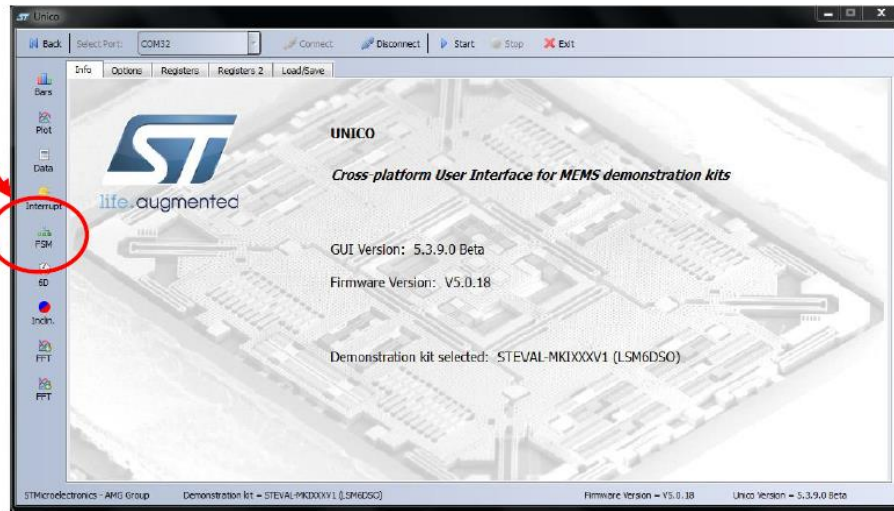
THRESHOLD
MASKS
TIMERS
EXECUTION
OUTPUT
VARIOUS

CONDITION		DESCRIPTION	CONDITION		DESCRIPTION
0x0	NOP	None execution - condition on current sample	0x8	LNTH2	Any triggered axis \leq THRS2
0x1	TI1	Timeout 1 expired	0x9	GLTH1	All triggered axis $>$ THRS1
0x2	TI2	Timeout 2 expired	0xA	LLTH1	All triggered axis \leq THRS1
0x3	TI3	Timeout 3 expired	0xB	GRTH1	Any triggered axis $>$ -THRS1
0x4	TI4	Timeout 4 expired	0xC	LRTH1	Any triggered axis \leq -THRS1
0x5	GNTH1	Any triggered axis $>$ THRS1	0xD	PZC	Any triggered axis crossed zero pos. slope
0x6	GNTH2	Any triggered axis $>$ THRS2	0xE	NZC	Any triggered axis crossed zero neg. slope
0x7	LNTH1	Any triggered axis \leq THRS1	0xF	N/A	N/A

- 复位条件总是先于继续执行条件的判断
- T1,T2,T3,T4,是定时器。
- GNTH1/2, LNTH1/2,GLTH1,LLTH1,GRTH1,LRTH1是由mask开关的和阈值进行比较的条件。
- PZC, NZC 是由mask开关的正向和负向过零的判断的条件指令。

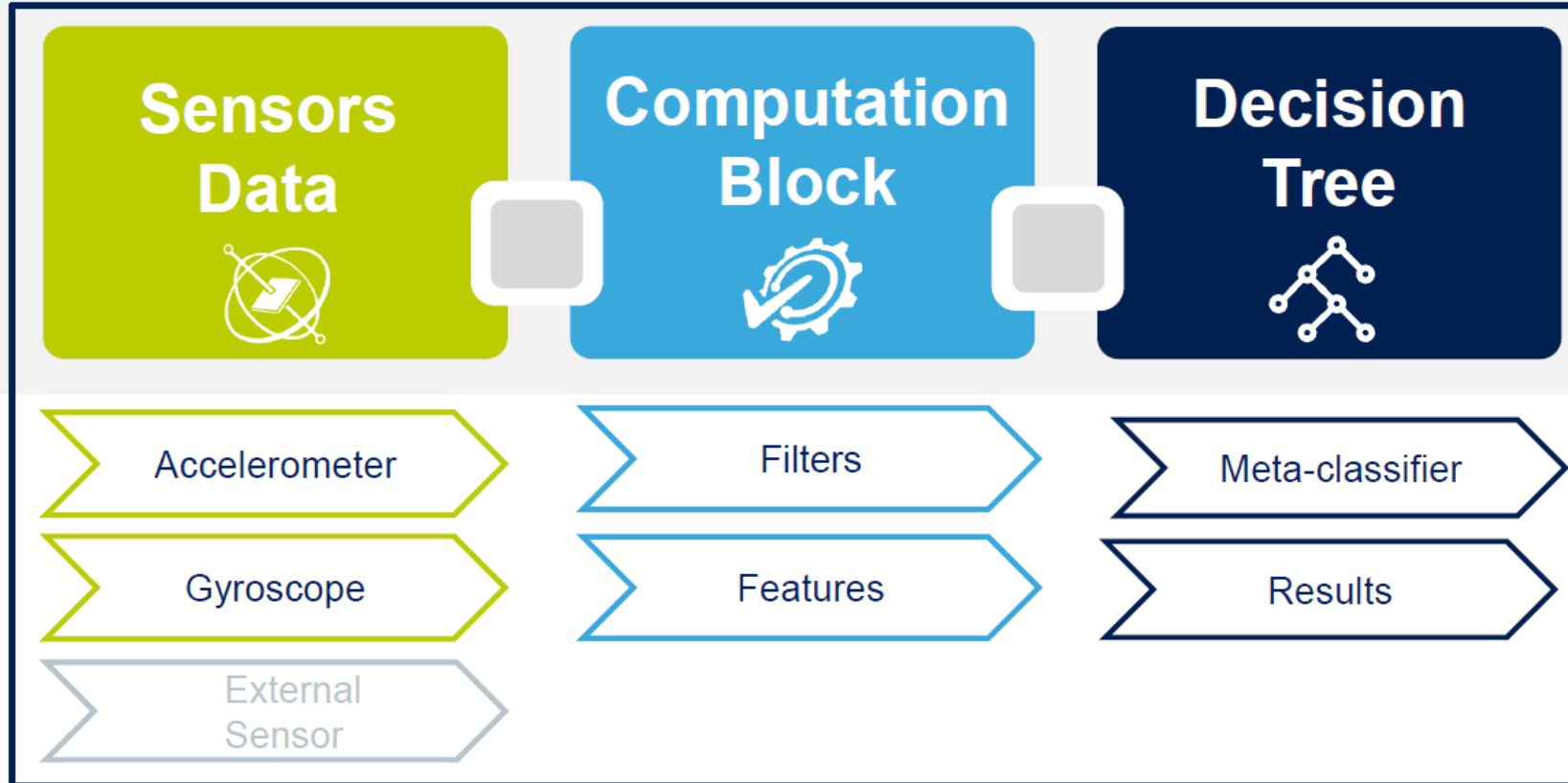
FSM 开发工具(UNICO)

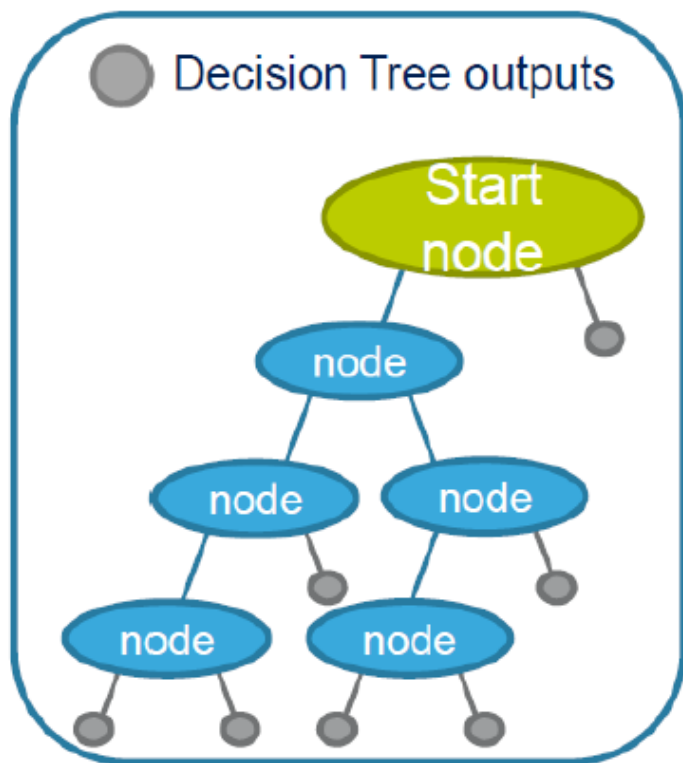
15



Machine Learning Core (MLC)

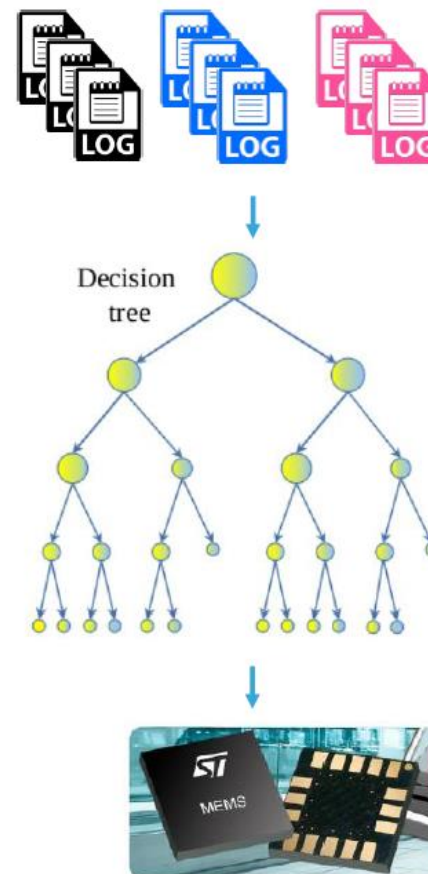
16





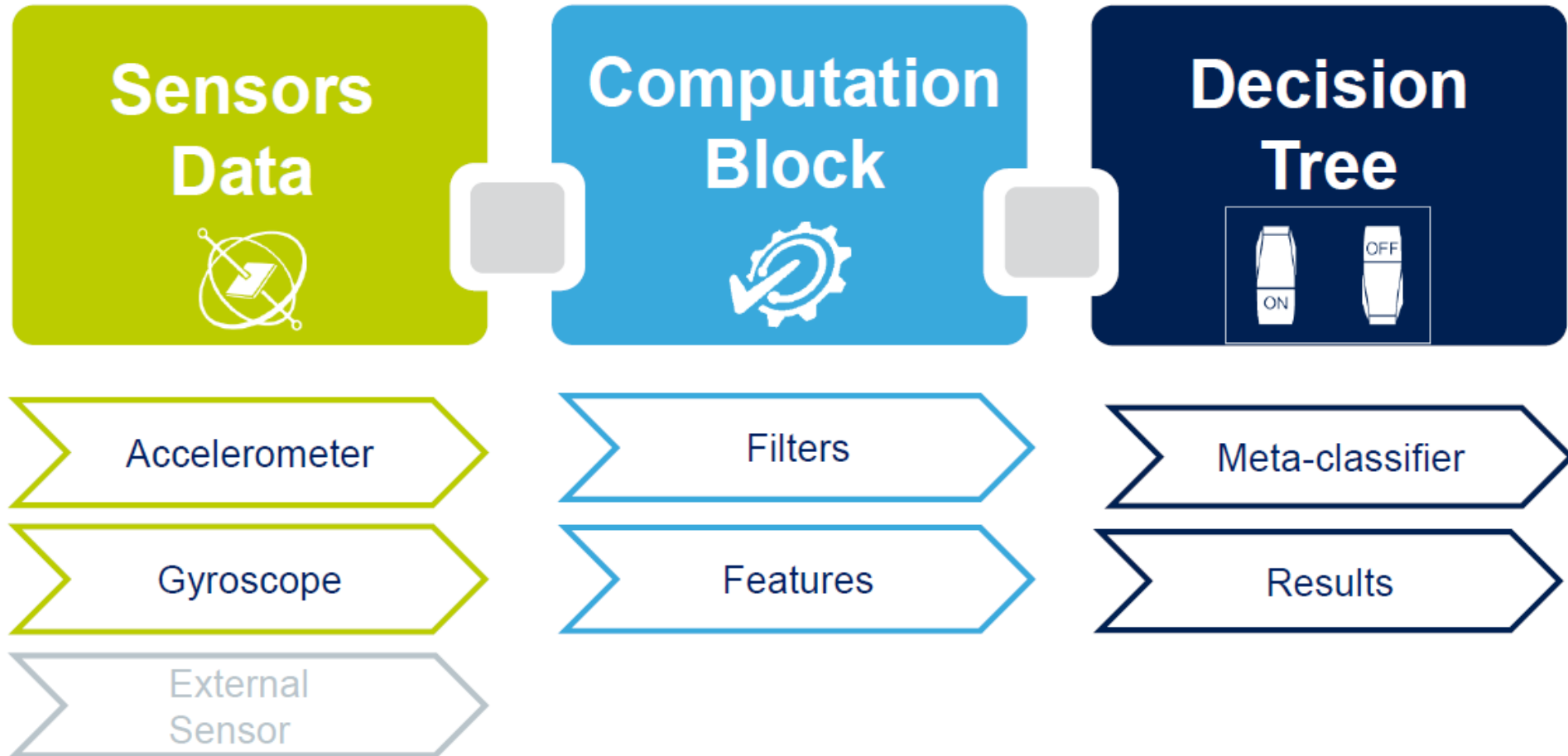
- Decision tree 是一种由一系列节点组成的数学工具
- 每个节点都可以生成基于阈值判断的 if – then-else 的条件
例如：
节点1：加速度x轴的平均值 $<500\text{mg}$
节点2：陀螺仪y轴方差值 $>100\text{dps}$
- Decision tree 执行的最终结果就是分辨出一系列可表征为不同的结果

- **Step1** -定义需要识别的内别
例如：活动，步行，慢跑，开车
- **Step2**- 对每个内别收集尽可能多数据。
例如：对同一个动作收集不同人的数据
- **Step3**-离线的数据分析
 - 对每个内别进行特征分析，选取合适的特征（如：平均值，方差值，能量，等等）
 - 应用机器学习工具（如：WEKA） 创建decision tree(阈值，节点，输出，等等)
- **Step4**-配置传感器，把decision tree的信息写入传感器。Decision tree 就可以低功耗运行在传感器里，减少MCU的功耗。



传感器数据处理流程

19



Sensors Data



Accelerometer

Gyroscope

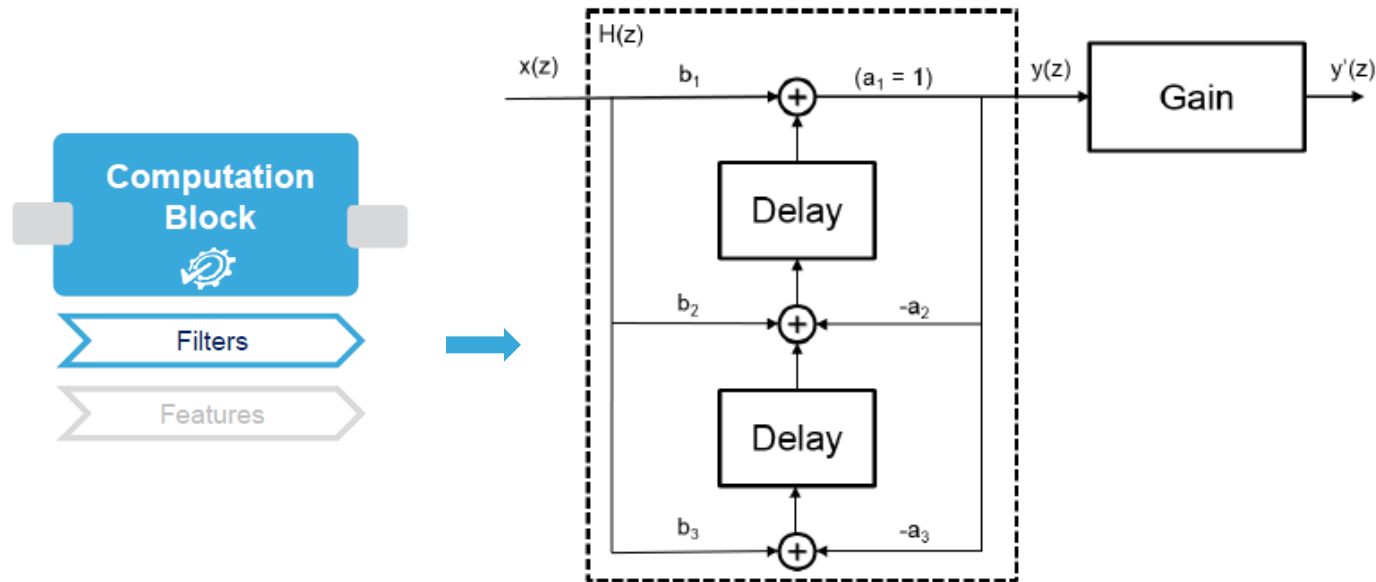
External Sensor

A wide Set of Inputs to be chosen

1. Accelerometer $[a_x \ a_y \ a_z]$, $[a_v]$, $[a_v^2]$
2. Gyroscope $[g_x \ g_y \ g_z]$, $[g_v]$, $[g_v^2]$
3. External sensor $[m_x \ m_y \ m_z]$, $[m_v]$, $[m_v^2]$

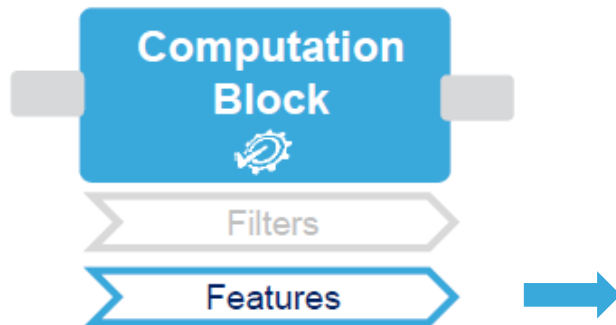
Magnitude Available

$$V = \sqrt{X^2 + Y^2 + Z^2}$$



传感器集成了2级IIR的滤波器，用户可选可以配置。

Filter	b_1	b_2	b_3	a_2	a_3	Gain
High pass	0.5	-0.5	0	0	0	1
Band pass	1	0	-1	configurable	configurable	configurable
IIR1	configurable	configurable	0	configurable	0	1
IIR2	configurable	configurable	configurable	configurable	configurable	1



特征：基于以下输入统计的特别因素

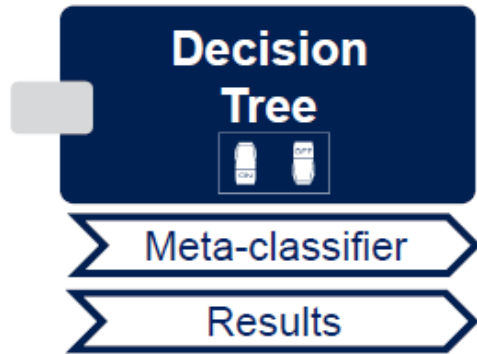
- 1) 输入数据（加速度，陀螺，等）
- 2) 对输入数据滤波（高通，带通，等）后的数据，

Feature Name	Feature Description
MEAN	Computes the average of the selected input in the defined time window $Mean = \frac{1}{WL} \sum_{k=0}^{WL-1} I_k$
VARIANCE	Computes the variance of the selected input in the defined time window $Variance = \left[\left(\frac{\sum_{WL} i^2}{WL} \right) - \left(\frac{\sum_{WL} i}{WL} \right)^2 \right]$
ENERGY	Computes the energy of the selected input in the defined time window $Energy = \sum_{k=0}^{WL} input^2$
PEAK TO PEAK	Computes the maximum peak to peak value of the selected input in the defined time window
ZERO CROSSING	Computes the number of times the selected input crosses a selected threshold in the defined time window
POSITIVE ZERO CROSSING	Computes the number of times the selected input crosses a selected threshold in the defined time window. Only transitions with positive slope are considered.
NEGATIVE ZERO CROSSING	Computes the number of times the selected input crosses a selected threshold in the defined time window. Only transitions with negative slopes are considered.
PEAK DETECTOR	Counts the number of peaks (positive and negative) of the selected input in the defined time window
POSITIVE PEAK DETECTOR	Counts the number of positive peaks of the selected input in the defined time window
NEGATIVE PEAK DETECTOR	Counts the number of negative peaks of the selected input in the defined time window
MINIMUM	Minimal value of the selected input in the defined time window
MAXIMUM	Maximum value of the selected input in the defined time window

WL = Window Length
(time window for features computation)

Meta-classifier 和结果

23

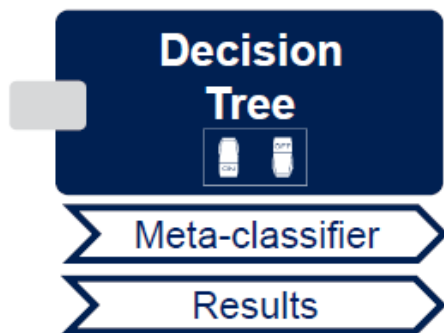


- 对每一个sample的处理，decision tree都会输出一个结果。
- Meta-classifier(可选)可以对decision tree 的输出进行滤波。
 - Meta-classifier 用一个内部计数器对decision tree的输出进行滤波
 - 只有当计数器等于最终的指定值时，meta-classifier 才会更新最终的结果

例如：

结果：A B。 A 指定的最终计数器的值是3，B是4。

Result before meta-classifier	A	A	A	B	A	B	B	B	A	B	B	B	A	A	A
Counter A	1	2	3	2	3	2	1	0	1	0	0	0	1	2	3
Counter B	0	0	0	1	0	1	2	3	2	3	4	5	4	3	2
Result after meta-classifier	X	X	A	A	A	A	A	A	A	A	B	B	B	B	A



Result	Register
Decision Tree 1	PRG_SENS_SRC0 (70h)
Decision Tree 2	PRG_SENS_SRC1 (71h)
Decision Tree 3	PRG_SENS_SRC2 (72h)
Decision Tree 4	PRG_SENS_SRC3 (73h)
Decision Tree 5	PRG_SENS_SRC4 (74h)
Decision Tree 6	PRG_SENS_SRC5 (75h)
Decision Tree 7	PRG_SENS_SRC6 (76h)
Decision Tree 8	PRG_SENS_SRC7 (77h)

- 每个decision tree 的结果都可以通过专门的寄存器来访问
 - PRG_SENS_SRCx (70-77h), 参考规格书
- 当decision tree的结果改变时可以产生一个中断。
 - 在以下寄存器可以访问中断标识
 - PROGSENS_STATUS_MAINPAGE(38h)
 - PROGSENS_STATUS(15h) [page1 of embedded advanced features registers]
 - 这个中断标志可以通过寄存器使能到硬件中断管脚1
 - PROGSENS_INT1(0Dh) [page1 of embedded advanced features registers]

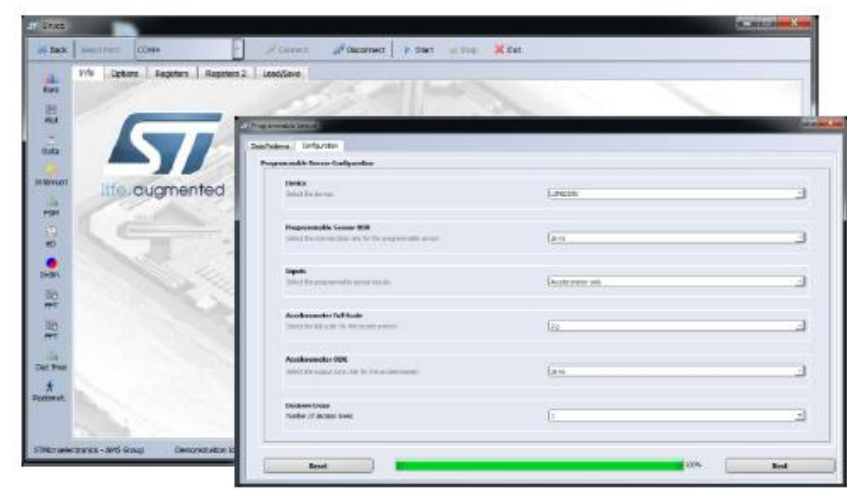
- 对于以下硬件资源，传感器提供了有限的支持
 - Decision tree 的数量
 - Decision tree的大小，主要指支持的最大节点数量
 - Decision tree 的输出结果数量
 - Meta-classifier
 - 特征处理单元和滤波器的数量

严格讲没有限制对于滤波器和特征的处理，但强烈建议减少了节点的数量，当配置了较多的滤波器和特征处理单元

	LSM6DSRX	LSM6DSOX
Maximum number of decision trees	8	8
Maximum number of Nodes (Total number for all the decision trees)	512 (*)	256 (*)
Maximum number of results per decision tree	256	16
Result sub-groups for meta-classifier	8	4
Number of Filters and Features	(*)	(*)

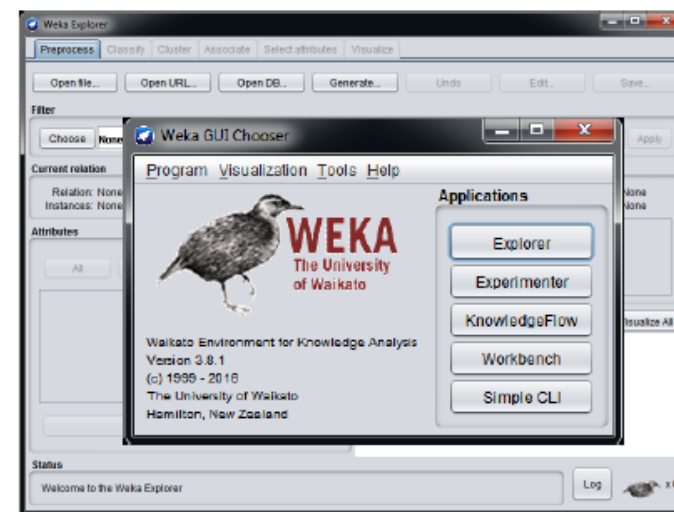
- **Unico GUI**

- 数据采集
- 传感器配置
- 传感器编程的配置
 - ✓ 数据分类管理
 - ✓ 传感器编程的设置（数据频率，量程，数据长度窗口，滤波器，特征处理单元，结果输出等。）
 - ✓ 滤波器和特征单元的计算，生成 ARFF文件和UCF文件
- 配置文件的装载（.ucf）
- Decision tree 输出的可视化显示以及记录



- **WEKA(机器学习的工具)**

- 属性选择（.arff文件）
- 数据滤波
- 生成decision tree
- Decision tree 性能的评估（节点数量，精度，等）



一 数据模型

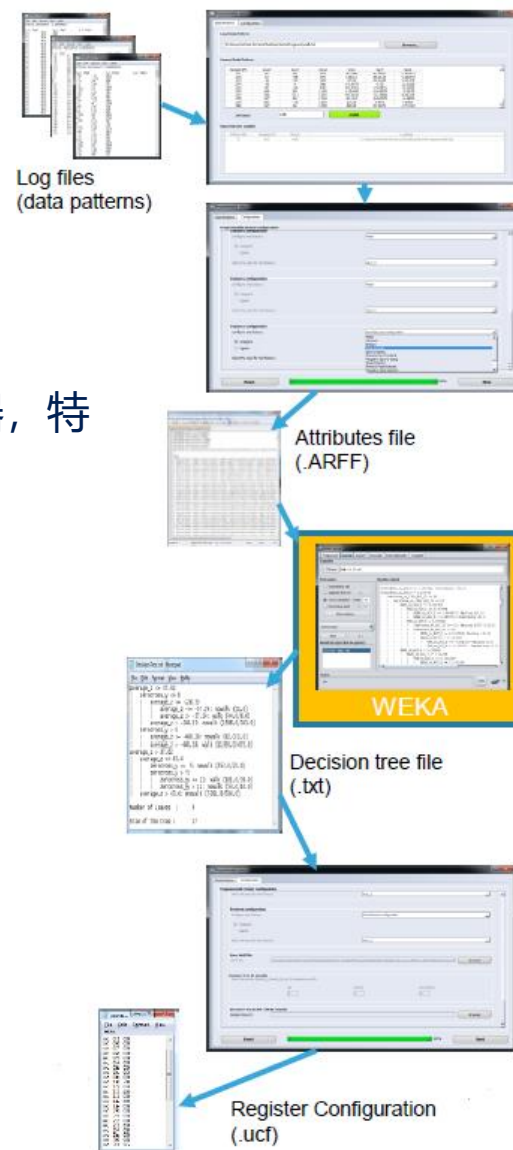
- 1) 打开Unico GUI工具, 选择传感器
- 2) 传感器一般配置
- 3) 数据采集保存文件

二 传感器的编程

- 1) 打开工具的MLC模块
- 2) 加载第一步采集的数据并按期待结果分类
- 3) 选择输入, 数据频率, 量程, 处理数据长度窗口, 滤波器, 特征单元选择, 等。
- 4) 生成ARFF 文件
- 5) 打开WEKA(机器学习工具)
 - i. 装载ARFF文件
 - ii. 选择属性
 - iii. 生成decision tree
 - iv. 保存decision tree到text文件
- 6) 在传感器编程模块装载decision tree的text 文件
- 7) 配置decision tree的输出值以及meta-classifier
- 8) 保存传感器所有寄存器设置到.ucf 文件。

三 器件配置

- 1) 在Unico 工具里装载.ucf 文件
- 2) 在Unico里查看decision tree的输出。

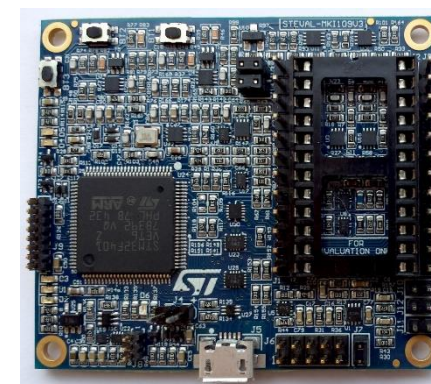


Lsm6dsox:

https://www.st.com/content/st_com/en/products/mems-and-sensors/inemo-inertial-modules/lsm6dsox.html

STEVAL-MKI109V3:

https://www.st.com/content/st_com/en/products/evaluation-tools/product-evaluation-tools/mems-motion-sensor-eval-boards/steval-mki109v3.html



Unico GUI:

https://www.st.com/content/st_com/en/products/embedded-software/evaluation-tool-software/stsw-mki109w.html





life.augmented