COSMIC 功能规模度量方法 **4.0.1** 版

电饭煲案例分析

2.0 版

2016 年 **3** 月

致谢

2.0 版的评审人员		
Alain Abran * École de Technologie Supérieure, Université du Québec Canada	Jean-Marc Desharnais École de Technologie Supérieure, Université du Québec Canada	Peter Fagg Pentad Ltd United Kingdom
Luigi Lavazza Universita degli Studi dell'Insubria Italy	Arlan Lesterhuis The Netherlands	Dylan Ren Measures Technology LLC China
Charles Symons * United Kingdom	Sylvie Trudel Université du Québec à Montréal Canada	Frank Vogelezang Ordina The Netherlands
Chris Woodward United Kingdom		

*标星号为本案例 2.0 版编辑人员

中文版贡献者	
翻译组织者	麦哲思科技（北京）有限公司 www.measures.net.cn 电话：400-1781727
初版翻译	张倩倩 麦哲思科技高级 QA
校对	郭玲 麦哲思科技高级咨询顾问，COSMIC 讲师

注：对 COSMIC 及本文档的任何疑问或指正之处，请加入 COSMIC 交流 QQ 群——309842452。

版本控制

下表为本案例分析的部分变更履历。

日期	评审者	修订/增加
2000	Abran, Desharnais, Fagg, Oligny, St-Pierre, Symons, De Tran-Cao	COSMIC-FFP 版本。软件定时控制
2003-01-02	Abran, O'Neill, Vinh Tuong Ho, et al …….	-使词汇与 ISO 19761:2003 同步; -明确哪些功能需求分配给硬件，哪些功能需求分配给软件； -将需求分离为 3 个迭代原型
2008-05-22	Abran, Fagg, O'Neill, Khelifi, Symons	使原型一符合 COSMIC v3.0, ISO 19761:2008.
2010-11-30	Abran, Symons	
2016-01-?	Abran. Lesterhuis, Symons, Trudel	重大修订。硬件和软件需求的清晰分离。与 COSMIC v4.0.1[1]一致，并考虑到 Lavazza[2]提出的一些观点。

目录

电饭煲的功能性需求

1.1 背景

这是某家制造公司生产的第一个电饭煲：该公司决定通过使用继承的原型进行产品开发。这里给出的案例分析对应于第一个要开发的原型，以及为第二个原型添加的其他可能的功能。在制造原型时获得的经验教训将有助于更好地决定是否应该将部分产品功能性需求重新分配给软件而不是硬件。

对于这个简单的电饭煲原型，系统工程师将决定：

- 哪些功能将通过硬件设备实现（第 1.2 章节）；
- 软硬件的交互（第 1.3 章节）；
- 哪些功能将通过软件实现（第 1.4 章节）。

请记住，这是一个原型，并没有明确描述一个可正常运作的电饭煲的所有功能。

因此，度量人员必须根据功能性需求中描述的功能进行度量，而不是其他可能存在的功能。

1.2 硬件功能

电源开关为所有硬件设备供电。该开关必须由电饭煲使用者开启，这样电饭煲才可以开始工作。

该电饭煲安装了以下硬件设备，可以直接向软件发送或接收数据。除了 “煮饭模式”按钮，它是将煮饭模式存放在一个随机访问存储器（“RAM”）中，使之可以被软件访问。

1. 煮饭模式按钮是由三个相互连接的按钮组成的，它允许使用者从三种煮饭模式（慢速，正常，快速）中选择一种。
   a) 当使用者使用者按下其中一个煮饭模式按钮时，所选的煮饭模式会被存入 RAM 中，供软件访问。
   b) 一旦按下启动按钮，硬件将不允许改变煮饭模式。
2. 启动按钮
   当使用者按下启动按钮：
   a) 向软件发送一个“开始”信号；
   b) 启动一个硬件定时器，为软件提供唯一的时间参考。如果使用者按下启动按钮而没有设置煮饭模式：
   c) 硬件将设置 RAM 中的煮饭模式为默认值“正常”。
3. 定时器向软件发出三种类型的信号。
   a) 每隔 1 秒，定时器会发出一个信号，通知运行时间 t 的值（即 t=0，t=1，t=2，t=3……；所消耗的时间分别是 0，1，2，3……）；
   b) 每隔 30 秒，定时器会发出一个信号，让软件更新电饭煲的目标温度。（此信号首先在 t=0 时发出，随后每隔 30 秒发出）；
   c) 每隔 5 秒，定时器会发出一个信号，让软件检查电饭煲的实际温度。（此信号首先在 t=5 时发出，随后每隔 5 秒发出）。
   当 t=0 时，信号按 b），a）顺序发出。
   在三种信号必须同时发出的情况下，按 b），a），c）顺序发出。
4. 在启动时，煮饭灯会收到软件发送的指令而开启。
5. 只读存储器（‘ROM’）用于存储图 2 中所示的“目标温度、模式、运行时间”数据，软件是可以访问这些数据的。根据运行的时间和煮饭模式，软件使用此表中的值来确定目标温度。

6. 收到软件的请求时，温度传感器进行测温，使软件可获取温度值。
7. 加热器收到软件发出的信号，进行开/关控制。
8. 在任何时候按下停止按钮，将会：
   a) 停止定时器（如果定时器重新启动，则运行时间信号将从 t=0 开始）；
   b) 切断所有设备的电源。

在这个电饭煲的原型中，按钮、电源和电饭煲盖子之间的安全交互是由硬件控制的，可以忽略。

图 1 显示了使用者所看到的电饭煲控制面板。

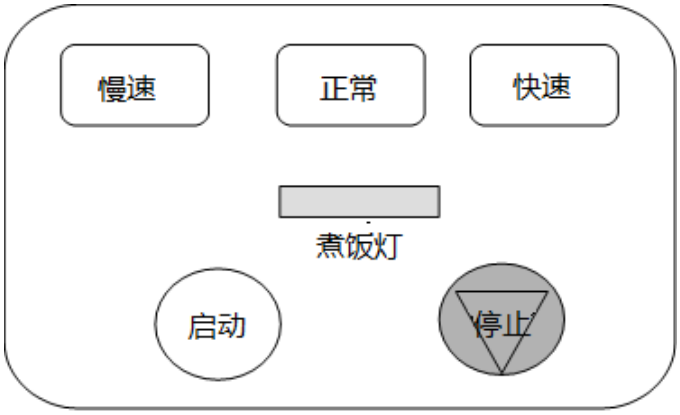

图 1.使用者控制面板

1.3 软、硬件的交互

软件必须：

1. 接收来自启动按钮的信号；
2. 接收来自定时器的三种不同类型的信号（每隔 5 秒和 30 秒的时钟信号，以及在一个 30 秒时钟信号之后的运行时间）；
3. 从 RAM 中获取煮饭模式；
4. 从 ROM 中获取“目标温度、模式、运行时间”数据；
5. 从温度传感器获取当前温度；
6. 向煮饭灯发送一个“打开”命令；
7. 向加热器发送一个“打开”或“关闭”命令；
8. 在 RAM 中存储当前目标温度，并从 RAM 中检索（或“获得”）目标温度。

1.4 软件功能性用户需求（FUR）

FUR1- 收到开始信号

软件：

a) 向煮饭灯发送一个“打开”信号；
b) 向加热器发送一个“打开”信号；

FUR2 - 收到 30 秒时钟信号

从 t=0 开始，每隔 30 秒，软件：

a) 接收 30 秒时钟信号；
b) 等待并接收运行时间的信号；
c) 从 RAM 中获取煮饭模式；
d) 在时间=[当前运行时间+30 秒]时，通过 ROM 中的数据，根据煮饭模式选择一个新的目标温度，见图 2；
e) 将这个新的目标温度存储到 RAM 中，这将成为当前的目标温度，直到下一个 30 秒时钟信号。

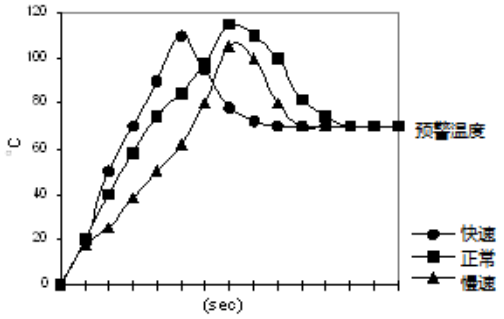

图 2-不同煮饭模式每隔 30 秒的目标温度

（水平轴=每隔 30 秒的运行时间）

FUR3 - 收到 5 秒时钟信号

从 t=5 秒时开始，每隔 5 秒，软件：

a) 接收 5 秒时钟信号；
b) 从 RAM 中获取当前目标温度；
c) 从温度传感器获取实际的传感器温度；
d) 如果“当前目标温度”大于“实际传感器温度”，则向加热器发送一个打开信号；否则，它会向加热器发送一个关闭信号。

基于本案例的特定目的，只通过软件实现明确分配给软件的功能。所有其他功能（包括那些尚未明确记录或没有要求的功能）都将通过第一个原型的硬件实现，也包括工程师在构建原型过程中可能发现的一些必需的功能。

度量策略

2.1 度量目的

度量目的是基于特定的功能性用户需求确定第一个原型的应用软件的规模。

2.2 度量范围

范围是功能性用户需求（FUR）中指定分配给软件的所有功能。FUR 没有区分软件的分解层级。电饭煲软件在应用层中。

2.3 识别功能用户

对于此电饭煲的第一个原型，软件有 5 个功能用户

- 定时器
- 启动按钮
- 温度传感器
- 煮饭灯
- 加热器

注意：RAM 和 ROM 并不是功能用户。（一个功能用户被定义为待度量软件的 FUR 中“数据的发送者或接收者”[1]）请参阅如下进一步说明。

电饭煲的使用者并不是软件的功能用户，使用者是作为产品的电饭煲的用户。使用者只是通过硬件与软件进行交互，硬件会向软件发送信号或从软件接收信号（也就是说，使用者通过煮饭模式、启动按钮和停止按钮与软件进行间接交互，并通过煮饭灯了解当前情况。）。

根据已知需求，图 3 展示了软件的环境图、功能用户和它们之间的边界。

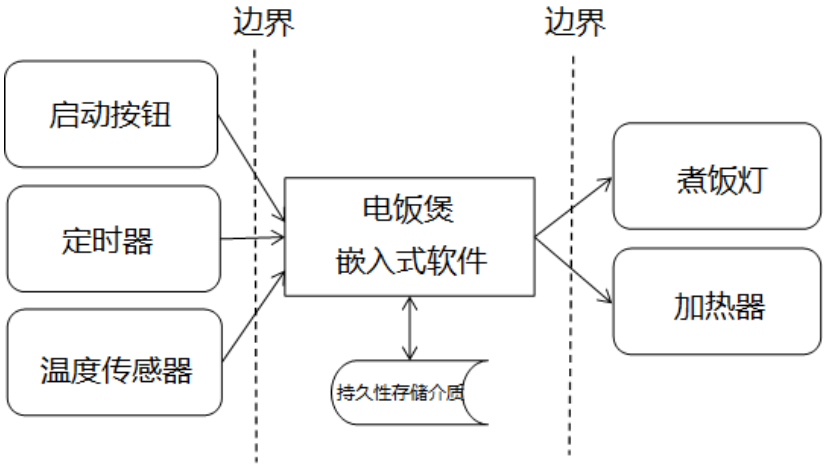

图 3-从功能性用户需求得到电饭煲软件的环境图

只向软件发送数据的三个硬件设备：

- 启动按钮
- 定时器
- 温度传感器。（按照 COSMIC 方法中关于“哑”传感器的规则[1]，软件的“请求发送数据”命令可以作为输入数据移动。）

只接收来自软件的数据的两个硬件设备：

- 煮饭灯
- 加热器

2.4 其他度量策略参数

颗粒度级别。电饭煲的 FUR 是“功能处理颗粒度级别”，因为功能用户是单独的硬件设备（不是这些设备的组合），且电饭煲软件必须响应的是单个事件（不是一组事件）。

持久性存储介质。COSMIC 模型不识别特定类型的硬件存储设备，例如 RAM 和 ROM。它只识别模型中的持久性存储介质，需要存储以下数据：

- 当前煮饭模式；
- 每种煮饭模式每隔 30 秒的目标温度（如图 2 所示）；
- 当前目标温度。

映射和度量阶段

3.1 识别软件的触发事件

从功能性用户需求中，可以确定以下触发事件：

1 按下启动按钮。（FUR1）

2 每隔 30 秒的定时器信号事件。（FUR2）

3 这个事件是由定时器创建的，当软件选择一个新的目标温度时，从 t=0 开始，每隔 30 秒触发。

4 每隔 5 秒的定时器信号事件。（FUR3）

这个事件是由定时器创建的，当软件需要重置加热器时，从 t=5 秒开始，每隔 5 秒触发。

这些事件都是触发事件，因为它们会触发功能用户（启动按钮或定时器）根据各自的信号生成一个数据组，从而触发功能处理。

运行时间信号的发送并不是软件的触发事件。尽管运行时间是由定时器发出的数据组，FUR 规定，只有在接收到一个 30 秒的定时器信号后，软件才会对发送的运行时间信号做处理。

软件在接收到 30 秒的时间信号后，才会注意到（即接收）运行时间信号，但是在其他的时候会忽略运行时间信号。因此不会对规模度量造成影响。

3.2 识别功能处理

根据前一章节的触发事件，从需求中识别以下候选功能处理：

1 开始煮饭。（收到开始信号）

2 更新目标温度。（30 秒定时器信号）

3 检查电饭煲的温度，根据需要打开或关闭加热器。（5 秒定时器信号）

3.3 识别兴趣对象

正如在度量手册[1]，第 3.3.5 章节中描述的，在实时软件系统中，物理硬件设备可以是功能用户，也可以是兴趣对象。实际上，硬件设备是与软件进行交互，以提供或接收关于自身的数据。因此，兴趣对象和它们各自的数据属性如下。

启动按钮：启动信号

煮饭灯：煮饭灯开、关命令

加热器：加热器开、关命令

定时器：当前运行时间信号、30 秒时钟信号、5 秒时钟信号

温度传感器：实际的传感器温度

目标设置：煮饭模式、运行时间、目标温度（如图 2 所示）

当前设置：当前煮饭模式、当前目标温度。

功能用户也可以是兴趣对象，因此，由温度传感器提供的温度应该被称为“实际的传感器温度”，但是这种说法在日常用语中是很奇怪的。（我们不会将家用温度计的读数称为“温度计温度”，而是指其环境的温度，比如“室温”。） 然而，在实时软件中，传感器的状态是传感器的属性，不是它所在环境的属性。

3.4 功能处理及其数据移动

本章节描述了三个功能处理及其数据移动。规模使用 COSMIC 定义的单位：1COSMIC 功能点=1 CFP
数据移动类型缩写为 E=输入，X=输出，R=读，W=写。

功能处理 1:开始煮饭
触发事件:按下启动按钮

功能用户	子处理	数据组	兴趣对象	数据移动类型	CFP
启动按钮	收到开始信号	开始信号	启动按钮	E	1
加热器	向加热器发送启动命令	加热器开关命令	加热器	X	1
煮饭灯	向煮饭灯发送启动命令	煮饭灯开关命令	煮饭灯	X	1
				总规模：3CFP	

功能处理 2：更新目标温度
触发事件：30 秒的时钟信号

功能用户	子处理	数据组	兴趣对象	数据移动类型	CFP
定时器	收到 30 秒时钟信号	30 秒时钟信号	定时器	E	1
定时器	收到当前运行时间信号	当前运行时间信号	定时器	E	1
	获取煮饭模式	当前煮饭模式	当前设置	R	1
	【当前运行时间+30 秒】时获取最新目标温度和煮饭模式。（替代当前温度）	最新目标温度	目标设置	R	1
	存储（最新）当前目标温度	当前目标温度	当前设置	W	1
				总规模：5CFP	

功能处理 3：检查电饭煲温度
触发事件：5 秒的时钟信号

功能用户	子处理	数据组	兴趣对象	数据移动类型	CFP

定时器	收到 5 秒时钟信号	5 秒时钟信号	定时器	E	1
	获取当前目标温度	当前目标温度	当前设置	R	1
温度传感器	获取实际传感器温度	实际传感器温度	温度传感器	E	1
	决定是否开启或关闭加热器	(无 – 这是数据运算)	-	-	-
加热器	向加热器发送开、关命令（按照需要）	加热器开、关信号	加热器	X	1
					总规模: 4 CFP

此电饭煲的第一个原型的软件功能总规模是它的三个功能处理规模之和，即：

3 CFP + 5 CFP + 4 CFP = 12 CFP.

第二个原型可能的需求

本章节描述了 3 种可能的需求，作为电饭煲第二个原型的新需求。每个需求对软件规模的影响必须单独度量。

4.1 停止功能

硬件需求

在第一个原型中，所有的停止功能完全是由硬件处理的。

对电饭煲需求的第一个可能的改变是，当按下停止按钮时，硬件将会：

a) 停止定时器；
b) 向软件发送一个“停止煮饭”的信号。然后，软件必须执行其他的停止功能，这些停止功能在第一个原型中是由硬件执行的。

此更改对其他需求的影响如下。

软硬件的交互

软件必须接受来自硬件的“停止”信号。

软件 FUR4 - 将一些停止功能分配给软件

在收到停止按钮的停止信号后：

a) 软件向加热器发送一个关闭命令；
b) 软件向煮饭灯发送一个关闭命令；
c) 软件停止执行，即它进入了自感应的等待状态。

功能用户

现在软件有了一个新的功能用户：一个向软件发送信号的停止按钮。图 4 展示了更新后的软件环境图。

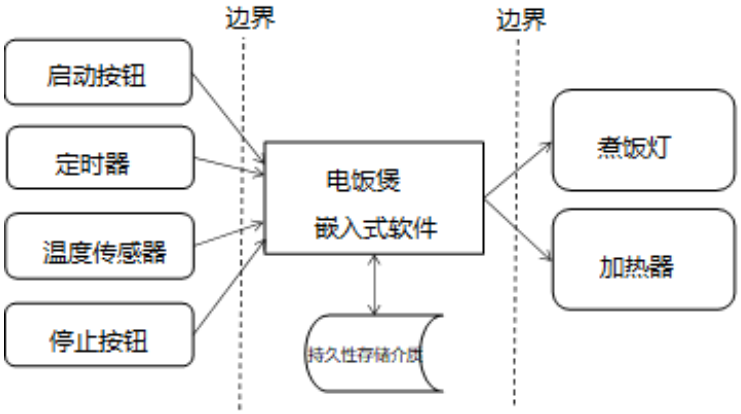

图 4-电饭煲软件第二个原型的环境图

触发事件-功能处理

使用者按下停止按钮触发的触发事件会额外触发一个 “停止煮饭”功能处理开始执行。

兴趣对象

现在多了一个兴趣对象，停止按钮，它有一个数据属性，停止按钮信号。

功能处理 4 及其数据移动

功能处理 4：停止煮饭 **触发事件：按下停止按钮**					
功能用户	**子处理**	**数据组**	**兴趣对象**	**数据移动类型**	**CFP**
停止按钮	接收到停止按钮信号	停止信号	停止按钮	E	1
加热器	向加热器发送停止命令	加热器开关命令	加热器	X	1
煮饭灯	向煮饭灯发送停止命令	煮饭灯开关命令	煮饭灯	X	1
	停止执行并进入等待状态	（无）	-	-	-
					总规模: 3 CFP

因此，该变更将使软件规模增加 3CFP，使原型 2 的软件总规模达到 15 CFP。

4.2 结合运行时间和 30 秒时钟信号

电饭煲需求的第二个可能的变更是，定时器将从 t = 0 开始，每隔 30 秒发出一个信号，该信号中也包括了运行时间，而不是如 1.2 章节中所描述的，分别发出运行时间和 30 秒的时间信号。

定时器将不再每隔 1 秒发出单独的运行时间信号。

根据第 1 章节的描述，需要对电饭煲需求做出以下修改。

硬件功能 3。现在变更为：

3. 定时器向软件发出两个信号：

a) 每隔 30 秒，会发出一个信号要求软件更新目标温度，并从 t=0 开始将当前运行时间也一并发送给软件。（此信号首先在 t=0 时发出，随后每隔 30 秒发出）。

b) 每隔 5 秒，会发出一个信号要求软件检查电饭煲的温度。（此信号首先在 t=5 时发出，随后每隔 5 秒发出）。

在两种类型的信号必须同时发出的情况下，按 a），b）顺序发出。

软、硬件交互 2。现在变更为：

2. 每隔 5 秒、30 秒，软件必须接收来自定时器的两种不同类型的信号。

功能用户。同第 2.3 章节。

软件触发事件。现在只有两个由定时器在每隔 30 秒和每隔 5 秒时产生触发事件。

兴趣对象。同第 3.3 章节。

软件 FUR。FUR2 的需求 a）现在变更为：

从 t=0 开始，每隔 30 秒，软件：

a) 接收 30 秒的信号和当前运行时间。

该变更对功能处理 2 及其数据移动的影响：

功能处理 2 现在每隔 30 秒只接收一个输入。这也表明从 t=0 开始的运行时间是作为第二个属性（而不是 30 秒时钟信号和运行时间信号两个独立输入）。

功能处理 2 的规模减少了 1CFP，为 4CFP。

变更后，原型 2 的软件规模从 12CFP 减少到 11CFP。

4.3 使用一个简单的定时器，每隔 1 秒产生一个时钟节拍

电饭煲需求的第三个变更是，第一个原型中使用的定时器（如第 1.2 章节中描述的，发出三个单独的信号，分别是运行时间，30 秒和 5 秒的信号）用一个简单的定时器取代，该定时器从 t=0 开始，每秒向软件发送一个节拍。其目的是让软件从 t=0 开始跟踪运行时间。

根据第 1 章节中的描述，需要对电饭煲需求做出以下修改。

硬件功能 3。现在变更为：

3. 定时器。从按下启动按钮后的 t=0 秒开始，每秒钟向软件发出一个简单的“节拍”信号（像时钟一样）。

软硬件交互 2。现在变更为：

2. 从按下启动按钮后的 t=0 秒开始，软件必须每隔 1 秒接收一个“节拍”信号。

功能用户。同第 2.3 章节图 3。

软件触发事件。旧定时器的三个触发事件现在被新的、更简单的定时器发出的触发事件所取代。

兴趣对象。同第 3.3 章节，除以下 2 点：

- 这个新的、简单的定时器仅有一个属性，即“节拍”。
- 软件必须跟踪运行时间。因此，还有另一个兴趣对象，我们称之为“运行时间”，它有一个属性“当前运行时间”。

软件 FUR。现在要求软件必须从 t=0 开始跟踪运行时间。这意味着必须对现有的 FUR 进行一些更改，如下。

功能处理 1。“开始煮饭”

这个现有的过程必须被修改，以便为当前运行时间存储一个 t=0 的数值。（注意，在目前情况下，软件必须生成这个值，它没有得到一个 t 的值或任何节拍。）

功能处理 5。“控制煮饭”

一个新的功能处理 5 必须替代现有的功能处理 2 和 3，FUR 如下。

FUR5-在收到定时器的节拍信号后

从 t=0 开始，每隔 1 秒，软件：

a) 接收定时器的“节拍”信号；
b) 从持久存储介质获取当前运行时‘t’；
c) 增加 1 秒到当前运行时间‘t’，并持久存储更新后的值，使之取代之前的运行时间；
d) 如果 t=0，或者 t =30 秒的整数倍，那么软件将会：
    i. 从持久存储介质获取煮饭模式；
    ii. 在时间=当前运行时间+30 秒时，通过从持久存储介质中读取相应的目标温度，为煮饭模式选择一个新的目标温度（参见图 1）；
    iii. 将这个新的目标温度存入持久存储介质中，这将成为当前的目标温度，直到下一次时间点到达 30 秒的整数倍。
e) 如果 t>0，并且 t=5 秒的整数倍，那么软件将会：
    i. 从持久存储介质获取当前目标温度（如果从步骤 d ii 还没有获知）
    ii. 从温度传感器获取实际传感器温度；
    iii. 如果“当前目标温度”大于“实际传感器温度”，向加热器发送一个打开信号，否则，发送一个关闭信号。

功能处理和数据移动

修改后的功能处理 1 和新的功能处理 5，及它们的数据移动如下所示：

修改后的功能处理 1：开始煮饭 **触发事件：开始信号**					
功能用户	**子处理**	**数据组**	**兴趣对象**	**数据移动类型**	**CFP**
启动按钮	收到开始信号	开始信号	启动按钮	E	1
加热器	向加热器发送启动命令	加热器开关命令	加热器	X	1
煮饭灯	向煮饭灯发送启动命令	煮饭灯开关命令	煮饭灯	X	1
	存储 t=0 时当前运行时间的值	当前运行时间	运行时间	W	1
					总规模: 4 CFP

功能处理 5：控制电饭煲 **触发事件：1 秒时钟节拍**					
功能用户	**子处理**	**数据组**	**兴趣对象**	**数据移动类型**	**CFP**
定时器	接收 1 秒节拍	1 秒信号	定时器	E	1

	从持久存储介质获取当前运行时间	当前运行时间	运行时间	R	1
	增加 1 秒到当前运行时间	（数据运算）			
	存储当前运行时间	当前运行时间	运行时间	W	1
	如果 t=0 或者 t=30 的整数倍时，获取煮饭模式；否则，跳过此步和接下来的两步	煮饭模式	当前设置	R	1
	获取当前模式下运行时间+30 秒的最新目标温度（这将取代当前的目标温度）	最新目标温度	目标设置	R	1
	存储（最新）当前目标温度	当前目标温度	当前设置	W	1
	如果 t=5 的整数倍，那么就得到当前的目标温度（如果尚未知晓）；否则，终止。	当前目标温度	当前设置	R	1
温度传感器	获取实际传感器温度	实际传感器温度	温度传感器	E	1
	决定加热器是否必须打开或关闭	（数据运算）	-	-	-
加热器	向加热器发送开关命令（如果需要）	加热器开关命令	加热器	X	1
					总规模: 9 CFP

该变更将导致原型 2 最终的软件总规模为 4+9=13CFP（较之原型 1 的 12CFP）

注意：在 t=0 和之后的每个 30 秒中，此功能处理 5 只需要执行 8 个数据移动（8CFP）。

- 在 t=5 秒和之后的每个 5 秒中，除了在 30 秒倍数时，它执行了 6CFP。
- 在其他时间，例如 t=1,2,3,4,6,7……，它只执行 3CFP，以更新运行时间。

然而，功能处理 5 总共需要 9CFP 满足其所有的 FUR。

参考文献

所有 COSMIC 发表的文章都可以从 www.cosmsizing.org 的知识库中免费下载。

[1] ‘Measurement Manual (The COSMIC Implementation Guide for ISO/IEC 19761: 2011)’, version 4.0.1, April 2015

[2] Lavazza, L., Del Bianco, V., ‘A case study in functional size measurement’; Rice Cooker case study revisited’, IWSM/Mensura conference 2009.

附录-变更请求和建议程序

COSMIC 度量实践委员会（MPC）非常愿意接受反馈与建议，以及对此指南的变更请求（如需要）。本附录展示了如何与 COSMIC MPC 联系。所有与 COSMIC MPC 的联系都应该通过电子邮件的方式发送到下面的地址：
mpc-chair@cosmic-sizing.org

非正式的反馈和建议

关于此指南的非正式建议和/或反馈，比如理解或应用 COSMIC 方法的任何困难、一般性改进的建议等，都可以通过电子邮件发送到上面的地址。您的邮件将会被登记，并且通常在收到后的两周内给予答复。度量实践委员会不保证对一般性意见给予答复。

正式的变更请求

本指南的读者如发现某些文字错误，或需要澄清的地方，或者一些文字需要调整，那么可以提交一个正式的变更请求（“CR”）。正式的 CR 会被登记，并在收到后的两周内给予答复。每个 CR 将分配一个序号，在 COSMIC MPC 的成员中循环传递，COSMIC MPC 是一个世界范围内 COSMIC 方法的专家组。他们的正常评审周期最少需要一个月，如果变更请求很难解决，可能需要更长的时间。评审的结果可能是 CR 被接受，或者拒绝，或者“未决定待进一步讨论”（例如本 CR 要依赖另一个 CR），结果会尽快地反馈给提交者。

只有包含了下面的所有信息，才会被作为一个正式的 CR 接受。

- 提交 CR 的人员姓名，职位和单位
- 提交人的详细联系方式
- 提交日期
- 关于 CR 的目的的总体陈述（如“需要改进文本……”）
- 实际需要变更、替换或删除的文字（或澄清引用出处）
- 建议增加或替换的文字
- 关于变更的必要性的充分解释

提交 CR 的表格可以从门户网站 www.cosmic-sizing.org 获得。

COSMIC MPC 对 CR 的评审结果，以及在哪个版本应用这个 CR（如果被接受了的话）的决定是最终的决定。

方法应用的问题

COSMIC MPC 抱歉不能回答与 COSMIC 方法应用相关的问题。有商业机构能够提供本方法的培训和顾问工作，或者支持工具。详细情况请查询 www.cosmic-sizing.org 网站。