N- Body Problem

Microprocessor Systems

Presented by:

Mohit Bansal

Samir Senapati Polobody

Motivation

- Understanding of
  - Arm Cortex M4 – Tiva C Launchpad
  - PC Serial Communication using any programming language.

- Real time computation and communication between a microprocessor and host

- Applications
  - Astrophysical applications which use adaptive time steps
  - N-body algorithm have numerous application in molecular dynamics and plasma physics

Problem Description

- Simulate N – body Problem

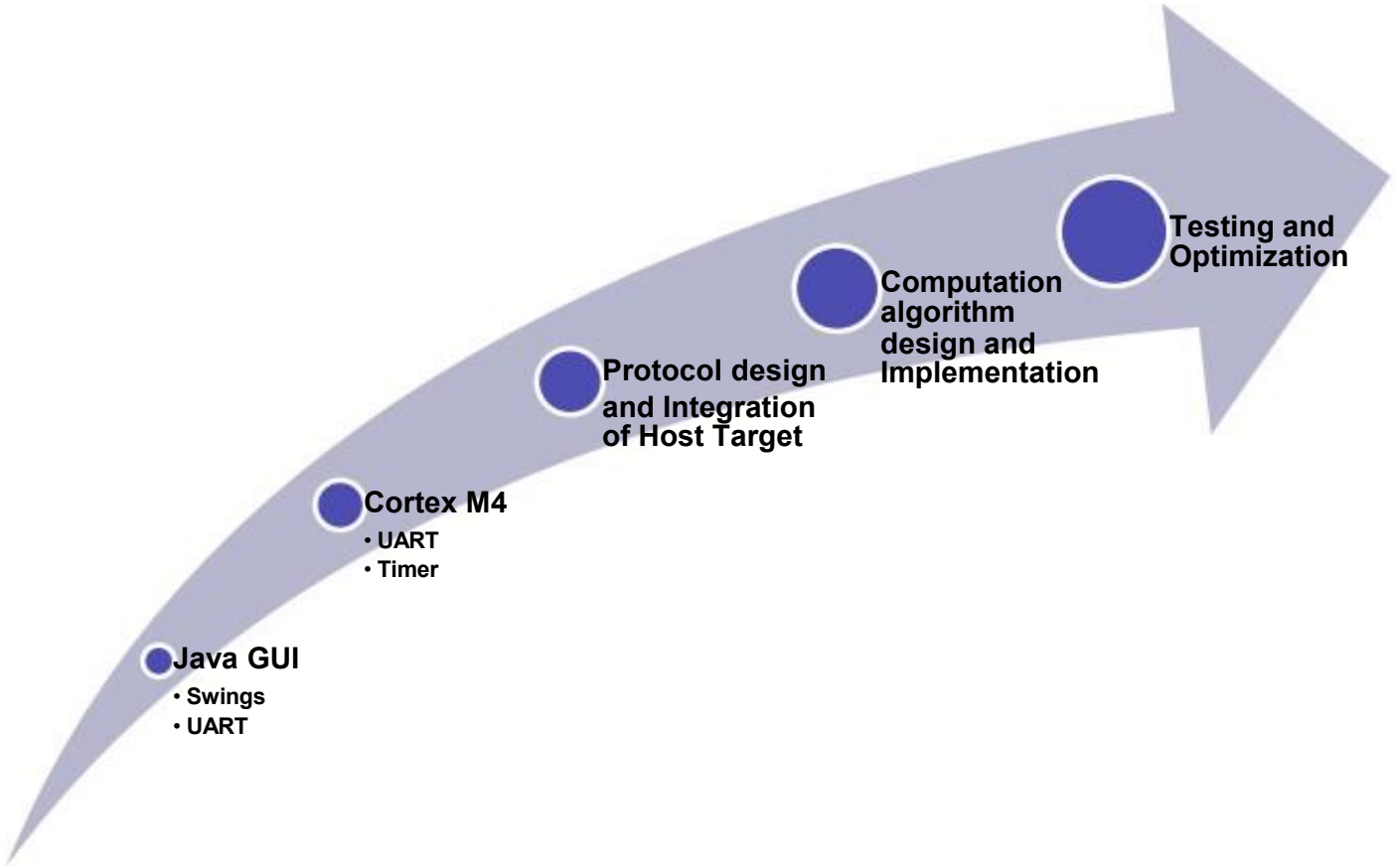
-Use the Launchpad to simulate the motion of a given set of planetary objects as governed by the effects of the forces of gravity on each other.

- ARM Cortex-M4 (Tiva C Launchpad) to be used to
  - –Two separate modes are to be made

    Computation mode

    Visualization mode

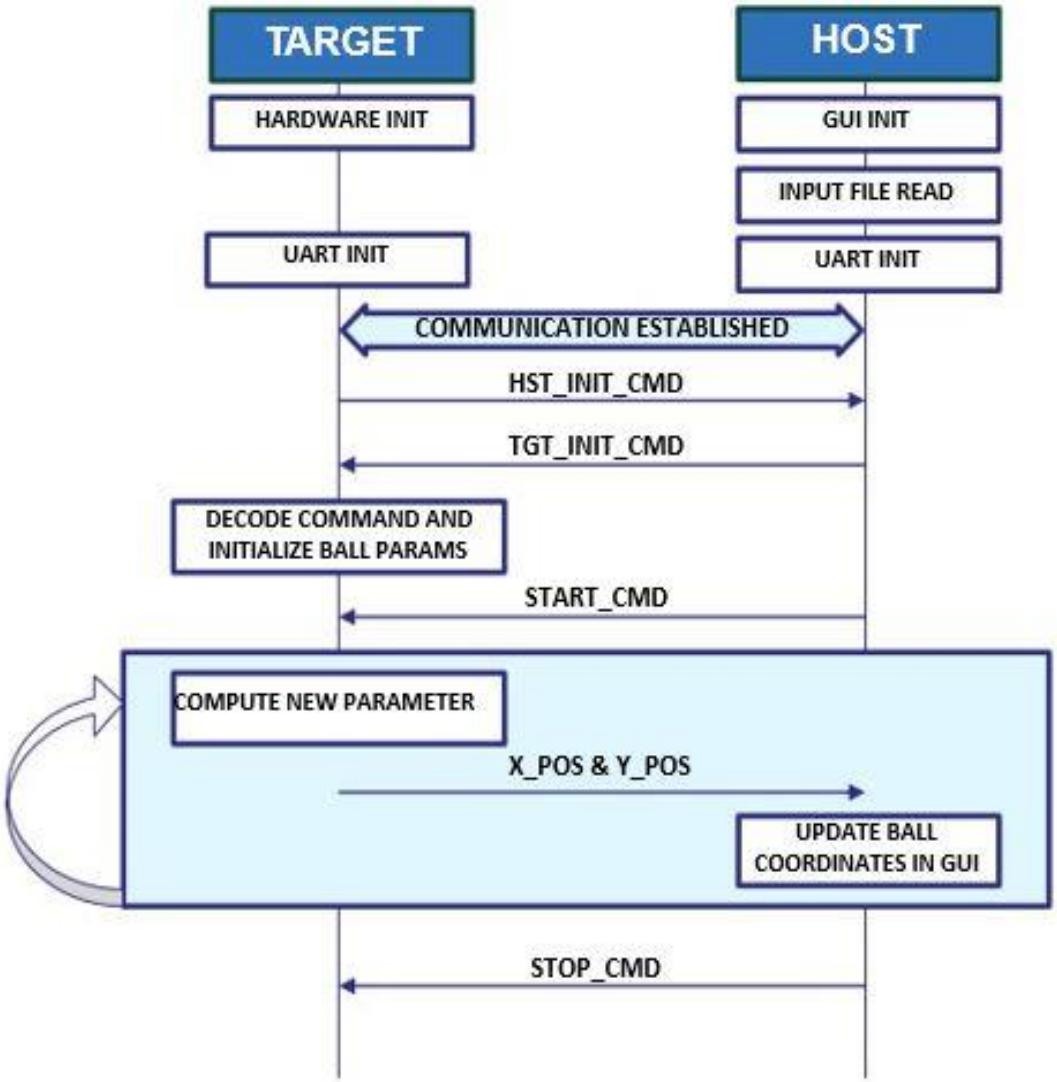
Technical Approach



Composition & Tasks

- GUI
  - Java Swing 8
- Microcontroller
  - ARM Cortex M-4
- Communication
  - UART

Technical Approach



Design – JAVA

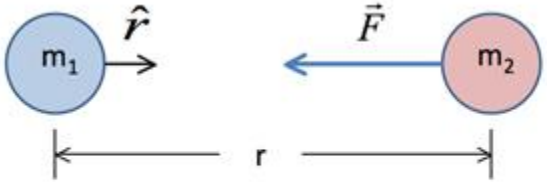
- Developed required skills required for GUI development in JAVA
- Develop method of integration of JAVA and Microcontroller.
- Develop Communication protocol between Microcontroller and JAVA
- Real time simulation of the ball according to the values sent by microcontroller

Design - Microcontroller

- Receive parameter from JAVA through input text file.
- Develop algorithm for the computation of N-Body Problem
- Real time computation of the position of the N-body based on the gravitational forces due to n bodies.
- Send the final computed coordinate values of the n bodies

Computation

Newton's law for Universal Gravitation

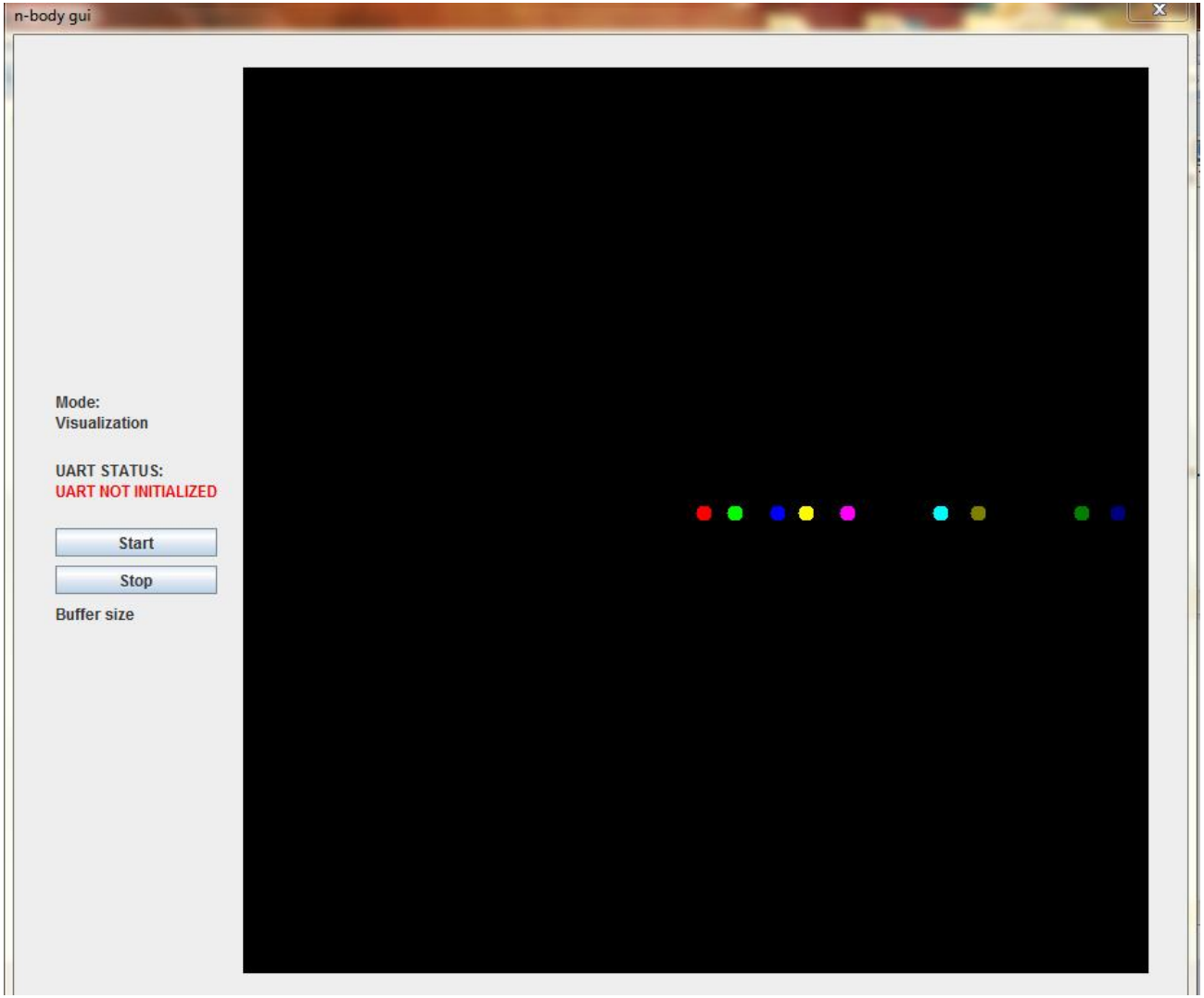


$$\vec{F} = -\frac{Gm_1m_2}{r^2}\hat{r}$$

Formula for computation of position of N-Body

$$\ddot{\mathbf{y}}_i = G \sum_{j=1, j\neq i}^{N} \frac{m_j(\mathbf{y}_j - \mathbf{y}_i)}{|\mathbf{y}_j - \mathbf{y}_i|^3},$$

Results


n-body gui
Mode:
Visualization
UART STATUS:
UART NOT INITIALIZED
Start
Stop
Buffer size

Challenges We Faced

- Communication
  - UART
  - Real Time Updating
- Collision algorithm
- Decoding of host and target commands
- Accuracy
- Making the design scalable for dynamic number of balls
- Optimization : Max number of balls


Thank You